Package: evd (via r-universe)

May 20, 2026

Version 2.3-7.1

Date 2024-04-23

Title Functions for Extreme Value Distributions

Author Alec Stephenson. Function fbvpot by Chris Ferro.

Maintainer Alec Stephenson <alec_stephenson@hotmail.com>

Imports stats, grDevices, graphics

Suggests interp

Description Extends simulation, distribution, quantile and density functions to univariate and multivariate parametric extreme value distributions, and provides fitting functions which calculate maximum likelihood estimates for univariate and bivariate maxima models, and for univariate and bivariate threshold models.

LazyData yes

License GPL-3

NeedsCompilation yes

Depends R (>= 2.10)

Repository https://ste6an-hub.r-universe.dev

Date/Publication 2024-09-22 03:02:17 UTC

RemoteUrl https://github.com/cran/evd

RemoteRef HEAD

RemoteSha d42be7d4de5bc5e0a0ee126bf4d31fce3b30de2c

Contents

---

abvevd *Parametric Dependence Functions of Bivariate Extreme Value Models*

Description

Calculate or plot the dependence function A for nine parametric bivariate extreme value models.

Usage

```
abvevd(x = 0.5, dep, asy = c(1,1), alpha, beta, model = c("log", "alog",
    "hr", "neglog", "aneglog", "bilog", "negbilog", "ct", "amix"),
     rev = FALSE, plot = FALSE, add = FALSE, lty = 1, lwd = 1, col = 1,
     blty = 3, blwd = 1, xlim = c(0,1), ylim = c(0.5,1), xlab = "t",
     ylab = "A(t)", ...)
```

Arguments

x	A vector of values at which the dependence function is evaluated (ignored if plot or add is TRUE). $A(1/2)$ is returned by default since it is often a useful summary of dependence.
dep	Dependence parameter for the logistic, asymmetric logistic, Husler-Reiss, negative logistic and asymmetric negative logistic models.
asy	A vector of length two, containing the two asymmetry parameters for the asymmetric logistic and asymmetric negative logistic models.
alpha, beta	Alpha and beta parameters for the bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models.
model	The specified model; a character string. Must be either "log" (the default), "alog", "hr", "neglog", "aneglog", "bilog", "negbilog", "ct" or "amix" (or any unique partial match), for the logistic, asymmetric logistic, Husler-Reiss, negative logistic, asymmetric negative logistic, bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models respectively. The definition of each model is given in rbvevd. If parameter arguments are given that do not correspond to the specified model those arguments are ignored, with a warning.
rev	Logical; reverse the dependence function? This is equivalent to evaluating the function at 1-x.
plot	Logical; if TRUE the function is plotted. The x and y values used to create the plot are returned invisibly. If plot and add are FALSE (the default), the arguments following add are ignored.

add	Logical; add to an existing plot? The existing plot should have been created using either abvevd or abvnonpar, the latter of which plots (or calculates) a non-parametric estimate of the dependence function.
lty, blty	Function and border line types. Set blty to zero to omit the border.
lwd, blwd	Function an border line widths.
col	Line colour.
xlim, ylim	x and y-axis limits.
xlab, ylab	x and y-axis labels.
...	Other high-level graphics parameters to be passed to plot.

Details

Any bivariate extreme value distribution can be written as

$$G(z_1, z_2) = \exp\left[-(y_1 + y_2)A\left(\frac{y_1}{y_1 + y_2}\right)\right]$$

for some function $A(\cdot)$ defined on $[0, 1]$, where

$$y_i = \{1 + s_i(z_i - a_i)/b_i\}^{-1/s_i}$$

for $1 + s_i(z_i - a_i)/b_i > 0$ and $i = 1, 2$, with the (generalized extreme value) marginal parameters given by (a_i, b_i, s_i), $b_i > 0$. If $s_i = 0$ then y_i is defined by continuity.

$A(\cdot)$ is called (by some authors) the dependence function. It follows that $A(0) = A(1) = 1$, and that $A(\cdot)$ is a convex function with $\max(x, 1 - x) \leq A(x) \leq 1$ for all $0 \leq x \leq 1$. The lower and upper limits of A are obtained under complete dependence and independence respectively. $A(\cdot)$ does not depend on the marginal parameters.

Some authors take B(x) = A(1-x) as the dependence function. If the argument rev = TRUE, then B(x) is plotted/evaluated.

Value

abvevd calculates or plots the dependence function for one of nine parametric bivariate extreme value models, at specified parameter values.

See Also

abvnonpar, fbvevd, rbvevd, amvevd

Examples

```
abvevd(dep = 2.7, model = "hr")
abvevd(seq(0,1,0.25), dep = 0.3, asy = c(.7,.9), model = "alog")
abvevd(alpha = 0.3, beta = 1.2, model = "negbi", plot = TRUE)

bvdata <- rbvevd(100, dep = 0.7, model = "log")
M1 <- fitted(fbvevd(bvdata, model = "log"))
abvevd(dep = M1["dep"], model = "log", plot = TRUE)
abvnonpar(data = bvdata, add = TRUE, lty = 2)
```

abvnonpar *Non-parametric Estimates for Dependence Functions of the Bivariate Extreme Value Distribution*

Description

Calculate or plot non-parametric estimates for the dependence function A of the bivariate extreme value distribution.

Usage

```
abvnonpar(x = 0.5, data, epmar = FALSE, nsloc1 = NULL,
    nsloc2 = NULL, method = c("cfg", "pickands", "tdo", "pot"),
     k = nrow(data)/4, convex = FALSE, rev = FALSE, madj = 0,
    kmar = NULL, plot = FALSE, add = FALSE, lty = 1, lwd = 1,
    col = 1, blty = 3, blwd = 1, xlim = c(0, 1), ylim = c(0.5, 1),
    xlab = "t", ylab = "A(t)", ...)
```

Arguments

x	A vector of values at which the dependence function is evaluated (ignored if plot or add is TRUE). $A(1/2)$ is returned by default since it is often a useful summary of dependence.
data	A matrix or data frame with two columns, which may contain missing values.
epmar	If TRUE, an empirical transformation of the marginals is performed in preference to marginal parametric GEV estimation, and the nsloc arguments are ignored.
nsloc1, nsloc2	A data frame with the same number of rows as data, for linear modelling of the location parameter on the first/second margin. The data frames are treated as covariate matrices, excluding the intercept. A numeric vector can be given as an alternative to a single column data frame.
method	The estimation method (see **Details**). Typically either "cfg" (the default) or "pickands". The method "tdo" performs poorly and is not recommended. The method "pot" is for peaks over threshold modelling where only large data values are used for estimation.
k	An integer parameter for the "pot" method. Only the largest k values are used, as described in bvtcplot.
convex	Logical; take the convex minorant?
rev	Logical; reverse the dependence function? This is equivalent to evaluating the function at 1-x.
madj	Performs marginal adjustments for the "pickands" method (see **Details**).
kmar	In the rare case that the marginal distributions are known, specifies the GEV parameters to be used instead of maximum likelihood estimates.
plot	Logical; if TRUE the function is plotted. The x and y values used to create the plot are returned invisibly. If plot and add are FALSE (the default), the arguments following add are ignored.

add	Logical; add to an existing plot? The existing plot should have been created using either abvnonpar or abvevd, the latter of which plots (or calculates) the dependence function for a number of parametric models.
lty, blty	Function and border line types. Set blty to zero to omit the border.
lwd, blwd	Function and border line widths.
col	Line colour.
xlim, ylim	x and y-axis limits.
xlab, ylab	x and y-axis labels.
...	Other high-level graphics parameters to be passed to plot.

Details

The dependence function $A(\cdot)$ of the bivariate extreme value distribution is defined in abvevd. Non-parametric estimates are constructed as follows. Suppose (z_{i1}, z_{i2}) for $i = 1, \ldots, n$ are n bivariate observations that are passed using the data argument. If epmar is FALSE (the default), then the marginal parameters of the GEV margins are estimated (under the assumption of independence) and the data is transformed using

$$y_{i1} = \{1 + \hat{s}_1(z_{i1} - \hat{a}_1)/\hat{b}_1\}_+^{-1/\hat{s}_1}$$

and

$$y_{i2} = \{1 + \hat{s}_2(z_{i2} - \hat{a}_2)/\hat{b}_2\}_+^{-1/\hat{s}_2}$$

for $i = 1, \ldots, n$, where $(\hat{a}_1, \hat{b}_1, \hat{s}_1)$ and $(\hat{a}_2, \hat{b}_2, \hat{s}_2)$ are the maximum likelihood estimates for the location, scale and shape parameters on the first and second margins. If nsloc1 or nsloc2 are given, the location parameters may depend on i (see fgev).

Two different estimators of the dependence function can be implemented. They are defined (on $0 \leq w \leq 1$) as follows.

method = "cfg" (Caperaa, Fougeres and Genest, 1997)

$$\log(A_c(w)) = \frac{1}{n}\left\{\sum_{i=1}^{n} \log(\max[(1-w)y_{i1}, wy_{i1}]) - (1-w)\sum_{i=1}^{n} y_{i1} - w\sum_{i=1}^{n} y_{i2}\right\}$$

method = "pickands" (Pickands, 1981)

$$A_p(w) = n\left\{\sum_{i=1}^{n} \min\left(\frac{y_{i1}}{w}, \frac{y_{i2}}{1-w}\right)\right\}^{-1}$$

Two variations on the estimator $A_p(\cdot)$ are also implemented. If the argument madj = 1, an adjustment given in Deheuvels (1991) is applied. If the argument madj = 2, an adjustment given in Hall and Tajvidi (2000) is applied. These are marginal adjustments; they are only useful when empirical marginal estimation is used.

Let $A_n(\cdot)$ be any estimator of $A(\cdot)$. None of the estimators satisfy $\max(w, 1-w) \leq A_n(w) \leq 1$ for all $0 \leq w \leq 1$. An obvious modification is

$$A_n^{'}(w) = \min(1, \max\{A_n(w), w, 1-w\}).$$

This modification is always implemented.

Convex estimators can be derived by taking the convex minorant, which can be achieved by setting convex to TRUE.

Value

abvnonpar calculates or plots a non-parametric estimate of the dependence function of the bivariate extreme value distribution.

Note

I have been asked to point out that Hall and Tajvidi (2000) suggest putting a constrained smoothing spline on their modified Pickands estimator, but this is not done here.

References

Caperaa, P. Fougeres, A.-L. and Genest, C. (1997) A non-parametric estimation procedure for bivariate extreme value copulas. *Biometrika*, **84**, 567–577.

Pickands, J. (1981) Multivariate extreme value distributions. *Proc. 43rd Sess. Int. Statist. Inst.*, **49**, 859–878.

Deheuvels, P. (1991) On the limiting behaviour of the Pickands estimator for bivariate extreme-value distributions. *Statist. Probab. Letters*, **12**, 429–439.

Hall, P. and Tajvidi, N. (2000) Distribution and dependence-function estimation for bivariate extreme-value distributions. *Bernoulli*, **6**, 835–844.

See Also

abvevd, amvnonpar, bvtcplot, fgev

Examples

```
bvdata <- rbvevd(100, dep = 0.7, model = "log")
abvnonpar(seq(0, 1, length = 10), data = bvdata, convex = TRUE)
abvnonpar(data = bvdata, method = "pick", plot = TRUE)

M1 <- fitted(fbvevd(bvdata, model = "log"))
abvevd(dep = M1["dep"], model = "log", plot = TRUE)
abvnonpar(data = bvdata, add = TRUE, lty = 2)
```

---

amvevd — *Parametric Dependence Functions of Multivariate Extreme Value Models*

Description

Calculate the dependence function A for the multivariate logistic and multivariate asymmetric logistic models; plot the estimated function in the trivariate case.

Usage

```
amvevd(x = rep(1/d,d), dep, asy, model = c("log", "alog"), d = 3, plot =
    FALSE, col = heat.colors(12), blty = 0, grid = if(blty) 150 else 50,
    lower = 1/3, ord = 1:3, lab = as.character(1:3), lcex = 1)
```

Arguments

x	A vector of length d or a matrix with d columns, in which case the dependence function is evaluated across the rows (ignored if plot is TRUE). The elements/rows of the vector/matrix should be positive and should sum to one, or else they should have a positive sum, in which case the rows are rescaled and a warning is given. $A(1/d, \ldots, 1/d)$ is returned by default since it is often a useful summary of dependence.
dep	The dependence parameter(s). For the logistic model, should be a single value. For the asymmetric logistic model, should be a vector of length $2^d - d - 1$, or a single value, in which case the value is used for each of the $2^d - d - 1$ parameters (see rmvevd).
asy	The asymmetry parameters for the asymmetric logistic model. Should be a list with $2^d - 1$ vector elements containing the asymmetry parameters for each separate component (see rmvevd and **Examples**).
model	The specified model; a character string. Must be either "log" (the default) or "alog" (or any unique partial match), for the logistic and asymmetric logistic models respectively. The definition of each model is given in rmvevd.
d	The dimension; an integer greater than or equal to two. The trivariate case d = 3 is the default.
plot	Logical; if TRUE, and the dimension d is three (the default dimension), the dependence function of a trivariate model is plotted. For plotting in the bivariate case, use abvevd. If FALSE (the default), the following arguments are ignored.
col	A list of colours (see image). The first colours in the list represent smaller values, and hence stronger dependence. Each colour represents an equally spaced interval between lower and one.
blty	The border line type, for the border that surrounds the triangular image. By default blty is zero, so no border is plotted. Plotting a border leads to (by default) an increase in grid (and hence computation time), to ensure that the image fits within it.
grid	For plotting, the function is evaluated at grid^2 points.
lower	The minimum value for which colours are plotted. By defualt lower $= 1/3$ as this is the theoretical minimum of the dependence function of the trivariate extreme value distribution.
ord	A vector of length three, which should be a permutation of the set $\{1, 2, 3\}$. The points $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ (the vertices of the simplex) are depicted clockwise from the top in the order defined by ord.The argument alters the way in which the function is plotted; it does not change the function definition.
lab	A character vector of length three, in which case the ith margin is labelled using the ith component, or NULL, in which case no labels are given. The actual location of the margins, and hence the labels, is defined by ord.
lcex	A numerical value giving the amount by which the labels should be scaled relative to the default. Ignored if lab is NULL.

Details

Let $z = (z_1, \ldots, z_d)$ and $w = (w_1, \ldots, w_d)$. Any multivariate extreme value distribution can be written as

$$G(z) = \exp\left\{-\left\{\sum_{j=1}^{d} y_j\right\} A\left(\frac{y_1}{\sum_{j=1}^{d} y_j}, \ldots, \frac{y_d}{\sum_{j=1}^{d} y_j}\right)\right\}$$

for some function A defined on the simplex $S_d = \{w \in R_+^d : \sum_{j=1}^{d} w_j = 1\}$, where

$$y_i = \{1 + s_i(z_i - a_i)/b_i\}^{-1/s_i}$$

for $1 + s_i(z_i - a_i)/b_i > 0$ and $i = 1, \ldots, d$, and where the (generalized extreme value) marginal parameters are given by (a_i, b_i, s_i), $b_i > 0$. If $s_i = 0$ then y_i is defined by continuity.

A is called (by some authors) the dependence function. It follows that $A(w) = 1$ when w is one of the d vertices of S_d, and that A is a convex function with $\max(w_1, \ldots, w_d) \leq A(w) \leq 1$ for all w in S_d. The lower and upper limits of A are obtained under complete dependence and mutual independence respectively. A does not depend on the marginal parameters.

Value

A numeric vector of values. If plotting, the smallest evaluated function value is returned invisibly.

See Also

amvnonpar, abvevd, rmvevd, image

Examples

```
amvevd(dep = 0.5, model = "log")
s3pts <- matrix(rexp(30), nrow = 10, ncol = 3)
s3pts <- s3pts/rowSums(s3pts)
amvevd(s3pts, dep = 0.5, model = "log")
## Not run: amvevd(dep = 0.05, model = "log", plot = TRUE, blty = 1)
amvevd(dep = 0.95, model = "log", plot = TRUE, lower = 0.94)

asy <- list(.4, .1, .6, c(.3,.2), c(.1,.1), c(.4,.1), c(.2,.3,.2))
amvevd(s3pts, dep = 0.15, asy = asy, model = "alog")
amvevd(dep = 0.15, asy = asy, model = "al", plot = TRUE, lower = 0.7)
```

---

amvnonpar *Non-parametric Estimates for Dependence Functions of the Multivariate Extreme Value Distribution*

Description

Calculate non-parametric estimates for the dependence function A of the multivariate extreme value distribution and plot the estimated function in the trivariate case.

Usage

```
amvnonpar(x = rep(1/d,d), data, d = 3, epmar = FALSE, nsloc = NULL,
    madj = 0, kmar = NULL, plot = FALSE, col = heat.colors(12),
    blty = 0, grid = if(blty) 150 else 50, lower = 1/3, ord = 1:3,
    lab = as.character(1:3), lcex = 1)
```

Arguments

x	A vector of length d or a matrix with d columns, in which case the dependence function is evaluated across the rows (ignored if plot is TRUE). The elements/rows of the vector/matrix should be positive and should sum to one, or else they should have a positive sum, in which case the rows are rescaled and a warning is given. $A(1/d, \ldots, 1/d)$ is returned by default since it is often a useful summary of dependence.
data	A matrix or data frame with d columns, which may contain missing values.
d	The dimension; an integer greater than or equal to two. The trivariate case d = 3 is the default.
epmar	If TRUE, an empirical transformation of the marginals is performed in preference to marginal parametric GEV estimation, and the nsloc argument is ignored.
nsloc	A data frame with the same number of rows as data, or a list containing d elements of this type, for linear modelling of the marginal location parameters. In the former case, the argument is applied to all margins. The data frames are treated as covariate matrices, excluding the intercept. Numeric vectors can be given as alternatives to single column data frames. A list can contain NULL elements for stationary modelling of selected margins.
madj	Performs marginal adjustments. See abvnonpar.
kmar	In the rare case that the marginal distributions are known, specifies the GEV parameters to be used instead of maximum likelihood estimates.
plot	Logical; if TRUE, and the dimension d is three (the default dimension), the dependence function of a trivariate extreme value distribution is plotted. For plotting in the bivariate case, use abvnonpar. If FALSE (the default), the following arguments are ignored.
col	A list of colours (see image). The first colours in the list represent smaller values, and hence stronger dependence. Each colour represents an equally spaced interval between lower and one.
blty	The border line type, for the border that surrounds the triangular image. By default blty is zero, so no border is plotted. Plotting a border leads to (by default) an increase in grid (and hence computation time), to ensure that the image fits within it.
grid	For plotting, the function is evaluated at grid^2 points.
lower	The minimum value for which colours are plotted. By default lower $= 1/3$ as this is the theoretical minimum of the dependence function of the trivariate extreme value distribution.

`ord` A vector of length three, which should be a permutation of the set $\{1, 2, 3\}$. The points $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ (the vertices of the simplex) are depicted clockwise from the top in the order defined by `ord`. The argument alters the way in which the function is plotted; it does not change the function definition.

`lab` A character vector of length three, in which case the `i`th margin is labelled using the `i`th component, or `NULL`, in which case no labels are given. By default, `lab` is `as.character(1:3)`. The actual location of the margins, and hence the labels, is defined by `ord`.

`lcex` A numerical value giving the amount by which the labels should be scaled relative to the default. Ignored if `lab` is `NULL`.

Value

A numeric vector of estimates. If plotting, the smallest evaluated estimate is returned invisibly.

Note

The rows of `data` that contain missing values are not used in the estimation of the dependence structure, but every non-missing value is used in estimating the margins.

The dependence function of the multivariate extreme value distribution is defined in `amvevd`. The function `amvevd` calculates and plots dependence functions of multivariate logistic and multivariate asymmetric logistic models.

The estimator plotted or calculated is a multivariate extension of the bivariate Pickands estimator defined in `abvnonpar`.

See Also

`amvevd`, `abvnonpar`, `fgev`

Examples

```
s5pts <- matrix(rexp(50), nrow = 10, ncol = 5)
s5pts <- s5pts/rowSums(s5pts)
sdat <- rmvevd(100, dep = 0.6, model = "log", d = 5)
amvnonpar(s5pts, sdat, d = 5)

## Not run: amvnonpar(data = sdat, plot = TRUE)
## Not run: amvnonpar(data = sdat, plot = TRUE, ord = c(2,3,1), lab = LETTERS[1:3])
## Not run: amvevd(dep = 0.6, model = "log", plot = TRUE)
## Not run: amvevd(dep = 0.6, model = "log", plot = TRUE, blty = 1)
```

anova.evd *Compare Nested EVD Objects*

Description

Compute an analysis of deviance table for two or more nested evd objects.

Usage

```
## S3 method for class 'evd'
anova(object, object2, ..., half = FALSE)
```

Arguments

object	An object of class "evd".
object2	An object of class "evd" that represents a model nested within object.
...	Further successively nested objects.
half	For some non-regular tesing problems the deviance difference is known to be one half of a chi-squared random variable. Set half to TRUE in these cases.

Value

An object of class c("anova", "data.frame"), with one row for each model, and the following five columns

M.Df	The number of parameters.
Deviance	The deviance.
Df	The number of parameters of the model in the previous row minus the number of parameters.
Chisq	The deviance minus the deviance of the model in the previous row (or twice this if half is TRUE).
Pr(>chisq)	The p-value calculated by comparing the quantile Chisq with a chi-squared distribution on Df degrees of freedom.

Warning

Circumstances may arise such that the asymptotic distribution of the test statistic is not chi-squared. In particular, this occurs when the smaller model is constrained at the edge of the parameter space. It is up to the user recognize this, and to interpret the output correctly.

In some cases the asymptotic distribution is known to be one half of a chi-squared; you can set half = TRUE in these cases.

See Also

fbvevd, fextreme, fgev, forder

Examples

```
uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
trend <- (-49:50)/100
M1 <- fgev(uvdata, nsloc = trend)
M2 <- fgev(uvdata)
M3 <- fgev(uvdata, shape = 0)
anova(M1, M2, M3)

bvdata <- rbvevd(100, dep = 0.75, model = "log")
M1 <- fbvevd(bvdata, model = "log")
M2 <- fbvevd(bvdata, model = "log", dep = 0.75)
M3 <- fbvevd(bvdata, model = "log", dep = 1)
anova(M1, M2)
anova(M1, M3, half = TRUE)
```

---

bvevd *Parametric Bivariate Extreme Value Distributions*

Description

Density function, distribution function and random generation for nine parametric bivariate extreme value models.

Usage

```
dbvevd(x, dep, asy = c(1, 1), alpha, beta, model = c("log", "alog",
    "hr", "neglog", "aneglog", "bilog", "negbilog", "ct", "amix"),
    mar1 = c(0, 1, 0), mar2 = mar1, log = FALSE)
pbvevd(q, dep, asy = c(1, 1), alpha, beta, model = c("log", "alog",
    "hr", "neglog", "aneglog", "bilog", "negbilog", "ct", "amix"),
    mar1 = c(0, 1, 0), mar2 = mar1, lower.tail = TRUE)
rbvevd(n, dep, asy = c(1, 1), alpha, beta, model = c("log", "alog",
    "hr", "neglog", "aneglog", "bilog", "negbilog", "ct", "amix"),
    mar1 = c(0, 1, 0), mar2 = mar1)
```

Arguments

x, q	A vector of length two or a matrix with two columns, in which case the density/distribution is evaluated across the rows.
n	Number of observations.
dep	Dependence parameter for the logistic, asymmetric logistic, Husler-Reiss, negative logistic and asymmetric negative logistic models.
asy	A vector of length two, containing the two asymmetry parameters for the asymmetric logistic and asymmetric negative logistic models.
alpha, beta	Alpha and beta parameters for the bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models.

model	The specified model; a character string. Must be either "log" (the default), "alog", "hr", "neglog", "aneglog", "bilog", "negbilog", "ct" or "amix" (or any unique partial match), for the logistic, asymmetric logistic, Husler-Reiss, negative logistic, asymmetric negative logistic, bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models respectively. If parameter arguments are given that do not correspond to the specified model those arguments are ignored, with a warning.
mar1, mar2	Vectors of length three containing marginal parameters, or matrices with three columns where each column represents a vector of values to be passed to the corresponding marginal parameter.
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default), the distribution function is returned; the survivor function is returned otherwise.

Details

Define

$$y_i = y_i(z_i) = \{1 + s_i(z_i - a_i)/b_i\}^{-1/s_i}$$

for $1 + s_i(z_i - a_i)/b_i > 0$ and $i = 1, 2$, where the marginal parameters are given by mari $= (a_i, b_i, s_i)$, $b_i > 0$. If $s_i = 0$ then y_i is defined by continuity.

In each of the bivariate distributions functions $G(z_1, z_2)$ given below, the univariate margins are generalized extreme value, so that $G(z_i) = \exp(-y_i)$ for $i = 1, 2$. If $1 + s_i(z_i - a_i)/b_i \leq 0$ for some $i = 1, 2$, the value z_i is either greater than the upper end point (if $s_i < 0$), or less than the lower end point (if $s_i > 0$), of the ith univariate marginal distribution.

model = "log" (Gumbel, 1960)

The bivariate logistic distribution function with parameter dep $= r$ is

$$G(z_1, z_2) = \exp\left[-(y_1^{1/r} + y_2^{1/r})^r\right]$$

where $0 < r \leq 1$. This is a special case of the bivariate asymmetric logistic model. Complete dependence is obtained in the limit as r approaches zero. Independence is obtained when $r = 1$.

model = "alog" (Tawn, 1988)

The bivariate asymmetric logistic distribution function with parameters dep $= r$ and asy $= (t_1, t_2)$ is

$$G(z_1, z_2) = \exp\left\{-(1 - t_1)y_1 - (1 - t_2)y_2 - [(t_1 y_1)^{1/r} + (t_2 y_2)^{1/r}]^r\right\}$$

where $0 < r \leq 1$ and $0 \leq t_1, t_2 \leq 1$. When $t_1 = t_2 = 1$ the asymmetric logistic model is equivalent to the logistic model. Independence is obtained when either $r = 1$, $t_1 = 0$ or $t_2 = 0$. Complete dependence is obtained in the limit when $t_1 = t_2 = 1$ and r approaches zero. Different limits occur when t_1 and t_2 are fixed and r approaches zero.

model = "hr" (Husler and Reiss, 1989)

The Husler-Reiss distribution function with parameter dep $= r$ is

$$G(z_1, z_2) = \exp\left(-y_1\Phi\{r^{-1} + \tfrac{1}{2}r[\log(y_1/y_2)]\} - y_2\Phi\{r^{-1} + \tfrac{1}{2}r[\log(y_2/y_1)]\}\right)$$

where $\Phi(\cdot)$ is the standard normal distribution function and $r > 0$. Independence is obtained in the limit as r approaches zero. Complete dependence is obtained as r tends to infinity.

`model = "neglog"` (Galambos, 1975)

The bivariate negative logistic distribution function with parameter `dep` $= r$ is

$$G(z_1, z_2) = \exp\left\{-y_1 - y_2 + [y_1^{-r} + y_2^{-r}]^{-1/r}\right\}$$

where $r > 0$. This is a special case of the bivariate asymmetric negative logistic model. Independence is obtained in the limit as r approaches zero. Complete dependence is obtained as r tends to infinity. The earliest reference to this model appears to be Galambos (1975, Section 4).

`model = "aneglog"` (Joe, 1990)

The bivariate asymmetric negative logistic distribution function with parameters parameters `dep` $= r$ and `asy` $= (t_1, t_2)$ is

$$G(z_1, z_2) = \exp\left\{-y_1 - y_2 + [(t_1 y_1)^{-r} + (t_2 y_2)^{-r}]^{-1/r}\right\}$$

where $r > 0$ and $0 < t_1, t_2 \leq 1$. When $t_1 = t_2 = 1$ the asymmetric negative logistic model is equivalent to the negative logistic model. Independence is obtained in the limit as either r, t_1 or t_2 approaches zero. Complete dependence is obtained in the limit when $t_1 = t_2 = 1$ and r tends to infinity. Different limits occur when t_1 and t_2 are fixed and r tends to infinity. The earliest reference to this model appears to be Joe (1990), who introduces a multivariate extreme value distribution which reduces to $G(z_1, z_2)$ in the bivariate case.

`model = "bilog"` (Smith, 1990)

The bilogistic distribution function with parameters `alpha` $= \alpha$ and `beta` $= \beta$ is

$$G(z_1, z_2) = \exp\left\{-y_1 q^{1-\alpha} - y_2(1-q)^{1-\beta}\right\}$$

where $q = q(y_1, y_2; \alpha, \beta)$ is the root of the equation

$$(1-\alpha)y_1(1-q)^{\beta} - (1-\beta)y_2 q^{\alpha} = 0,$$

$0 < \alpha, \beta < 1$. When $\alpha = \beta$ the bilogistic model is equivalent to the logistic model with dependence parameter `dep` $= \alpha = \beta$. Complete dependence is obtained in the limit as $\alpha = \beta$ approaches zero. Independence is obtained as $\alpha = \beta$ approaches one, and when one of α, β is fixed and the other approaches one. Different limits occur when one of α, β is fixed and the other approaches zero. A bilogistic model is fitted in Smith (1990), where it appears to have been first introduced.

`model = "negbilog"` (Coles and Tawn, 1994)

The negative bilogistic distribution function with parameters `alpha` $= \alpha$ and `beta` $= \beta$ is

$$G(z_1, z_2) = \exp\left\{-y_1 - y_2 + y_1 q^{1+\alpha} + y_2(1-q)^{1+\beta}\right\}$$

where $q = q(y_1, y_2; \alpha, \beta)$ is the root of the equation

$$(1+\alpha)y_1 q^{\alpha} - (1+\beta)y_2(1-q)^{\beta} = 0,$$

$\alpha > 0$ and $\beta > 0$. When $\alpha = \beta$ the negative bilogistic model is equivalent to the negative logistic model with dependence parameter `dep` $= 1/\alpha = 1/\beta$. Complete dependence is obtained in the limit as $\alpha = \beta$ approaches zero. Independence is obtained as $\alpha = \beta$ tends to infinity, and when one of α, β is fixed and the other tends to infinity. Different limits occur when one of α, β is fixed and the other approaches zero.

`model = "ct"` (Coles and Tawn, 1991)

The Coles-Tawn distribution function with parameters `alpha` $= \alpha > 0$ and `beta` $= \beta > 0$ is

$$G(z_1, z_2) = \exp\{-y_1[1 - \text{Be}(q; \alpha + 1, \beta)] - y_2 \text{Be}(q; \alpha, \beta + 1)\}$$

where $q = \alpha y_2/(\alpha y_2 + \beta y_1)$ and $\text{Be}(q; \alpha, \beta)$ is the beta distribution function evaluated at q with `shape1` $= \alpha$ and `shape2` $= \beta$. Complete dependence is obtained in the limit as $\alpha = \beta$ tends to infinity. Independence is obtained as $\alpha = \beta$ approaches zero, and when one of α, β is fixed and the other approaches zero. Different limits occur when one of α, β is fixed and the other tends to infinity.

`model = "amix"` (Tawn, 1988)

The asymmetric mixed distribution function with parameters `alpha` $= \alpha$ and `beta` $= \beta$ has a dependence function with the following cubic polynomial form.

$$A(t) = 1 - (\alpha + \beta)t + \alpha t^2 + \beta t^3$$

where α and $\alpha+3\beta$ are non-negative, and where $\alpha+\beta$ and $\alpha+2\beta$ are less than or equal to one. These constraints imply that beta lies in the interval [-0.5,0.5] and that alpha lies in the interval [0,1.5], though alpha can only be greater than one if beta is negative. The strength of dependence increases for increasing alpha (for fixed beta). Complete dependence cannot be obtained. Independence is obtained when both parameters are zero. For the definition of a dependence function, see abvevd.

Value

`dbvevd` gives the density function, `pbvevd` gives the distribution function and `rbvevd` generates random deviates, for one of nine parametric bivariate extreme value models.

Note

The logistic and asymmetric logistic models respectively are simulated using bivariate versions of Algorithms 1.1 and 1.2 in Stephenson(2003). All other models are simulated using a root finding algorithm to simulate from the conditional distributions.

The simulation of the bilogistic and negative bilogistic models requires a root finding algorithm to evaluate q within the root finding algorithm used to simulate from the conditional distributions. The generation of bilogistic and negative bilogistic random deviates is therefore relatively slow (about 2.8 seconds per 1000 random vectors on a 450MHz PIII, 512Mb RAM).

The bilogistic and negative bilogistic models can be represented under a single model, using the integral of the maximum of two beta distributions (Joe, 1997).

The Coles-Tawn model is called the Dirichelet model in Coles and Tawn (1991).

References

Coles, S. G. and Tawn, J. A. (1991) Modelling extreme multivariate events. *J. Roy. Statist. Soc., B*, **53**, 377–392.

Coles, S. G. and Tawn, J. A. (1994) Statistical methods for multivariate extremes: an application to structural design (with discussion). *Appl. Statist.*, **43**, 1–48.

Galambos, J. (1975) Order statistics of samples from multivariate distributions. *J. Amer. Statist. Assoc.*, **70**, 674–680.

Gumbel, E. J. (1960) Distributions des valeurs extremes en plusieurs dimensions. *Publ. Inst. Statist. Univ. Paris*, **9**, 171–173.

Husler, J. and Reiss, R.-D. (1989) Maxima of normal random vectors: between independence and complete dependence. *Statist. Probab. Letters*, **7**, 283–286.

Joe, H. (1990) Families of min-stable multivariate exponential and multivariate extreme value distributions. *Statist. Probab. Letters*, **9**, 75–81.

Joe, H. (1997) *Multivariate Models and Dependence Concepts*, London: Chapman & Hall.

Smith, R. L. (1990) Extreme value theory. In *Handbook of Applicable Mathematics* (ed. W. Ledermann), vol. 7. Chichester: John Wiley, pp. 437–471.

Stephenson, A. G. (2003) Simulating multivariate extreme value distributions of logistic type. *Extremes*, **6**(1), 49–60.

Tawn, J. A. (1988) Bivariate extreme value theory: models and estimation. *Biometrika*, **75**, 397–415.

See Also

abvevd, rgev, rmvevd

Examples

```
pbvevd(matrix(rep(0:4,2), ncol=2), dep = 0.7, model = "log")
pbvevd(c(2,2), dep = 0.7, asy = c(0.6,0.8), model = "alog")
pbvevd(c(1,1), dep = 1.7, model = "hr")

margins <- cbind(0, 1, seq(-0.5,0.5,0.1))
rbvevd(11, dep = 1.7, model = "hr", mar1 = margins)
rbvevd(10, dep = 1.2, model = "neglog", mar1 = c(10, 1, 1))
rbvevd(10, alpha = 0.7, beta = 0.52, model = "bilog")

dbvevd(c(0,0), dep = 1.2, asy = c(0.5,0.9), model = "aneglog")
dbvevd(c(0,0), alpha = 0.75, beta = 0.5, model = "ct", log = TRUE)
dbvevd(c(0,0), alpha = 0.7, beta = 1.52, model = "negbilog")
```

---

bvtcplot *Bivariate Threshold Choice Plot*

Description

Produces a diagnostic plot to assist with threshold choice for bivariate data.

Usage

```
bvtcplot(x, spectral = FALSE, xlab, ylab, ...)
```

Arguments

x	A matrix or data frame, ordinarily with two columns, which may contain missing values.
spectral	If TRUE, an estimate of the spectral measure is plotted instead of the diagnostic plot.
ylab, xlab	Graphics parameters.
...	Other arguments to be passed to the plotting function.

Details

If spectral is FALSE (the default), produces a threshold choice plot as illustrated in Beirlant et al. (2004). With n non-missing bivariate observations, the integers $k = 1, \ldots, n-1$ are plotted against the values $(k/n)r_{(n-k)}$, where $r_{(n-k)}$ is the $(n-k)$th order statistic of the sum of the margins following empirical transformation to standard Frechet.

A vertical line is drawn at k0, where k0 is the largest integer for which the y-axis is above the value two. If spectral is FALSE, the largest k0 data points are used to plot an estimate of the spectal measure $H([0, w])$ versus w.

Value

A list is invisibly returned giving k0 and the values used to produce the plot.

References

Beirlant, J., Goegebeur, Y., Segers, J. and Teugels, J. L. (2004) *Statistics of Extremes: Theory and Applications.*, Chichester, England: John Wiley and Sons.

See Also

fbvpot, tcplot

Examples

```
## Not run: bvtcplot(lossalae)
## Not run: bvtcplot(lossalae, spectral = TRUE)
```

---

ccbvevd *Calculate Conditional Copulas for Parametric Bivariate Extreme Value Distributions*

Description

Conditional copula functions, conditioning on either margin, for nine parametric bivariate extreme value models.

Usage

```
ccbvevd(x, mar = 2, dep, asy = c(1, 1), alpha, beta, model = c("log",
    "alog", "hr", "neglog", "aneglog", "bilog", "negbilog", "ct",
    "amix"), lower.tail = TRUE)
```

Arguments

x	A matrix or data frame, ordinarily with two columns, which may contain missing values. A data frame may also contain a third column of mode logical, which itself may contain missing values (see **Details**).
mar	One or two; conditions on this margin.
dep	Dependence parameter for the logistic, asymmetric logistic, Husler-Reiss, negative logistic and asymmetric negative logistic models.
asy	A vector of length two, containing the two asymmetry parameters for the asymmetric logistic and asymmetric negative logistic models.
alpha, beta	Alpha and beta parameters for the bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models.
model	The specified model; a character string. Must be either "log" (the default), "alog", "hr", "neglog", "aneglog", "bilog", "negbilog", "ct" or "amix" (or any unique partial match), for the logistic, asymmetric logistic, Husler-Reiss, negative logistic, asymmetric negative logistic, bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models respectively. If parameter arguments are given that do not correspond to the specified model those arguments are ignored, with a warning.
lower.tail	Logical; if TRUE (default), the conditional distribution function is returned; the conditional survivor function is returned otherwise.

Details

The function calculates $P(U_1 < x_1 | U_2 = x_2)$, where (U_1, U_2) is a random vector with Uniform(0,1) margins and with a dependence structure given by the specified parametric model. By default, the values of x_1 and x_1 are given by the first and second columns of the argument x. If mar = 1 then this is reversed.

If x has a third column x_3 of mode logical, then the function returns $P(U_1 < x_1 | U_2 = x_2, I = x_3)$, according to inference proceedures derived by Stephenson and Tawn (2004). See fbvevd. This requires numerical integration, and hence will be slower.

This function is mainly for internal use. It is used by plot.bvevd to calculate the conditional P-P plotting diagnostics.

Value

A numeric vector of probabilities.

References

Stephenson, A. G. and Tawn, J. A. (2004) Exploiting Occurence Times in Likelihood Inference for Componentwise Maxima. *Biometrika* **92**(1), 213–217.

See Also

rbvevd, fbvevd, plot.bvevd

---

chiplot *Dependence Measure Plots*

Description

Plots of estimates of the dependence measures chi and chi-bar for bivariate data.

Usage

```
chiplot(data, nq = 100, qlim = NULL, which = 1:2, conf = 0.95, trunc =
    TRUE, spcases = FALSE, lty = 1, cilty = 2, col = 1, cicol = 1,
    xlim = c(0,1), ylim1 = c(-1,1), ylim2 = c(-1,1), main1 = "Chi Plot",
    main2 = "Chi Bar Plot", xlab = "Quantile", ylab1 = "Chi", ylab2 =
    "Chi Bar", ask = nb.fig < length(which) && dev.interactive(), ...)
```

Arguments

data	A matrix or data frame with two columns. Rows (observations) with missing values are stripped from the data before any computations are performed.
nq	The number of quantiles at which the measures are evaluated.
qlim	The limits of the quantiles at which the measures are evaluated (see **Details**).
which	If only one plot is required, specify 1 for chi and 2 for chi-bar.
conf	The confidence coefficient of the plotted confidence intervals.
trunc	Logical; truncate the estimates at their theoretical upper and lower bounds?
spcases	If TRUE, plots greyed lines corresponding to the special cases of perfect positive/negative dependence and exact independence.
lty, cilty	Line types for the estimates of the measures and for the confidence intervals respectively. Use zero to supress.
col, cicol	Colour types for the estimates of the measures and for the confidence intervals respectively.
xlim, xlab	Limits and labels for the x-axis; they apply to both plots.
ylim1	Limits for the y-axis of the chi plot. If this is NULL (the default) the upper limit is one, and the lower limit is the minimum of zero and the smallest plotted value.
ylim2	Limits for the y-axis of the chi-bar plot.
main1, main2	The plot titles for the chi and chi-bar plots respectively.
ylab1, ylab2	The y-axis labels for the chi and chi-bar plots respectively.
ask	Logical; if TRUE, the user is asked before each plot.
...	Other arguments to be passed to matplot.

Details

These measures are explained in full detail in Coles, Heffernan and Tawn (1999). A brief treatment is also given in Section 8.4 of Coles(2001). A short summary is given as follows. We assume that the data are *iid* random vectors with common bivariate distribution function G, and we define the random vector (X, Y) to be distributed according to G.

The chi plot is a plot of q against empirical estimates of

$$\chi(q) = 2 - \log(\Pr(F_X(X) < q, F_Y(Y) < q))/\log(q)$$

where F_X and F_Y are the marginal distribution functions, and where q is in the interval (0,1). The quantity $\chi(q)$ is bounded by

$$2 - \log(2u - 1)/\log(u) \leq \chi(q) \leq 1$$

where the lower bound is interpreted as `-Inf` for $q \leq 1/2$ and zero for $q = 1$. These bounds are reflected in the corresponding estimates.

The chi bar plot is a plot of q against empirical estimates of

$$\bar{\chi}(q) = 2\log(1 - q)/\log(\Pr(F_X(X) > q, F_Y(Y) > q)) - 1$$

where F_X and F_Y are the marginal distribution functions, and where q is in the interval (0,1). The quantity $\bar{\chi}(q)$ is bounded by $-1 \leq \bar{\chi}(q) \leq 1$ and these bounds are reflected in the corresponding estimates.

Note that the empirical estimators for $\chi(q)$ and $\bar{\chi}(q)$ are undefined near $q = 0$ and $q = 1$. By default the function takes the limits of q so that the plots depicts all values at which the estimators are defined. This can be overridden by the argument `qlim`, which must represent a subset of the default values (and these can be determined using the component `quantile` of the invisibly returned list; see **Value**).

The confidence intervals within the plot assume that observations are independent, and that the marginal distributions are estimated exactly. The intervals are constructed using the delta method; this may lead to poor interval estimates near $q = 0$ and $q = 1$.

The function $\chi(q)$ can be interpreted as a quantile dependent measure of dependence. In particular, the sign of $\chi(q)$ determines whether the variables are positively or negatively associated at quantile level q. By definition, variables are said to be asymptotically independent when $\chi(1)$ (defined in the limit) is zero. For independent variables, $\chi(q) = 0$ for all q in (0,1). For perfectly dependent variables, $\chi(q) = 1$ for all q in (0,1). For bivariate extreme value distributions, $\chi(q) = 2(1 - A(1/2))$ for all q in (0,1), where A is the dependence function, as defined in `abvevd`. If a bivariate threshold model is to be fitted (using `fbvpot`), this plot can therefore act as a threshold identification plot, since e.g. the use of 95% marginal quantiles as threshold values implies that $\chi(q)$ should be approximately constant above $q = 0.95$.

The function $\bar{\chi}(q)$ can again be interpreted as a quantile dependent measure of dependence; it is most useful within the class of asymptotically independent variables. For asymptotically dependent variables (i.e. those for which $\chi(1) < 1$), we have $\bar{\chi}(1) = 1$, where $\bar{\chi}(1)$ is again defined in the limit. For asymptotically independent variables, $\bar{\chi}(1)$ provides a measure that increases with dependence strength. For independent variables $\bar{\chi}(q) = 0$ for all q in (0,1), and hence $\bar{\chi}(1) = 0$.

Value

A list with components `quantile`, `chi` (if `1` is in `which`) and `chibar` (if `2` is in `which`) is invisibly returned. The components `quantile` and `chi` contain those objects that were passed to the formal arguments `x` and `y` of `matplot` in order to create the chi plot. The components `quantile` and `chibar` contain those objects that were passed to the formal arguments `x` and `y` of `matplot` in order to create the chi-bar plot.

Author(s)

Jan Heffernan and Alec Stephenson

References

Coles, S. G., Heffernan, J. and Tawn, J. A. (1999) Dependence measures for extreme value analyses. *Extremes*, **2**, 339–365.

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values*, London: Springer–Verlag.

See Also

`fbvevd`, `fbvpot`, `matplot`

Examples

```
par(mfrow = c(1,2))
smdat1 <- rbvevd(1000, dep = 0.6, model = "log")
smdat2 <- rbvevd(1000, dep = 1, model = "log")
chiplot(smdat1)
chiplot(smdat2)
```

`clusters` *Identify Clusters of Exceedences*

Description

Identify clusters of exceedences.

Usage

```
clusters(data, u, r = 1, ulow = -Inf, rlow = 1, cmax = FALSE, keep.names
    = TRUE, plot = FALSE, xdata = seq(along = data), lvals = TRUE, lty =
    1, lwd = 1, pch = par("pch"), col = if(n > 250) NULL else "grey",
    xlab = "Index", ylab = "Data", ...)
```

Arguments

data	A numeric vector, which may contain missing values.
u	A single value giving the threshold, unless a time varying threshold is used, in which case u should be a vector of thresholds, typically with the same length as data (or else the usual recycling rules are applied).
r	A postive integer denoting the clustering interval length. By default the interval length is one.
ulow	A single value giving the lower threshold, unless a time varying lower threshold is used, in which case ulow should be a vector of lower thresholds, typically with the same length as data (or else the usual recycling rules are applied). By default there is no lower threshold (or equivalently, the lower threshold is -Inf).
rlow	A postive integer denoting the lower clustering interval length. By default the interval length is one.
cmax	Logical; if FALSE (the default), a list containing the clusters of exceedences is returned. If TRUE a numeric vector containing the cluster maxima is returned.
keep.names	Logical; if FALSE, the function makes no attempt to retain the names/indices of the observations within the returned object. If data contains a large number of observations, this can make the function run much faster. The argument is mainly designed for internal use.
plot	Logical; if TRUE a plot is given that depicts the identified clusters, and the clusters (if cmax is FALSE) or cluster maxima (if cmax is TRUE) are returned invisibly. If FALSE (the default), the following arguments are ignored.
xdata	A numeric vector with the same length as data, giving the values to be plotted on the x-axis.
lvals	Logical; should the values below the threshold and the line depicting the lower threshold be plotted?
lty, lwd	Line type and width for the lines depicting the threshold and the lower threshold.
pch	Plotting character.
col	Strips of colour col are used to identify the clusters. An observation is contained in the cluster if the centre of the corresponding plotting character is contained in the coloured strip. If col is NULL the strips are omitted. By default the strips are coloured "grey", but are omitted whenever data contains more than 250 observations.
xlab, ylab	Labels for the x and y axis.
...	Other graphics parameters.

Details

The clusters of exceedences are identified as follows. The first exceedence of the threshold initiates the first cluster. The first cluster then remains active until either r consecutive values fall below (or are equal to) the threshold, or until rlow consecutive values fall below (or are equal to) the lower threshold. The next exceedence of the threshold (if it exists) then initiates the second cluster, and so on. Missing values are allowed, in which case they are treated as falling below (or equal to) the threshold, but falling above the lower threshold.

Value

If cmax is FALSE (the default), a list with one component for each identified cluster. If cmax is TRUE, a numeric vector containing the cluster maxima. In any case, the returned object has an attribute acs, giving the average cluster size (where the cluster size is defined as the number of exceedences within a cluster), which will be NaN if there are no values above the threshold (and hence no clusters).

If plot is TRUE, the list of clusters, or vector of cluster maxima, is returned invisibly.

See Also

exi, exiplot

Examples

```
clusters(portpirie, 4.2, 3)
clusters(portpirie, 4.2, 3, cmax = TRUE)
clusters(portpirie, 4.2, 3, 3.8, plot = TRUE)
clusters(portpirie, 4.2, 3, 3.8, plot = TRUE, lvals = FALSE)
tvu <- c(rep(4.2, 20), rep(4.1, 25), rep(4.2, 20))
clusters(portpirie, tvu, 3, plot = TRUE)
```

---

confint.evd *Calculate Confidence Intervals*

Description

Calculate profile and Wald confidence intervals of parameters in fitted models.

Usage

```
## S3 method for class 'evd'
confint(object, parm, level = 0.95, ...)
## S3 method for class 'profile.evd'
confint(object, parm, level = 0.95, ...)
```

Arguments

object	Either a fitted model object (of class evd) for Wald confidence intervals, or a profile trace (of class profile.evd) for profile likelihood confidence intervals.
parm	A character vector of parameters; a confidence interval is calculated for each parameter. If missing, then intervals are returned for all parameters in the fitted model or profile trace.
level	A single number giving the confidence level.
...	Not used.

Value

A matrix with two columns giving lower and upper confidence limits.

For profile confidence intervals, this function assumes that the profile trace is unimodal. If the profile trace is not unimodal then the function will give spurious results.

See Also

profile.evd

Examples

```
m1 <- fgev(portpirie)
confint(m1)
## Not run: pm1 <- profile(m1)
## Not run: plot(pm1)
## Not run: confint(pm1)
```

---

evind.test *Perform Hypothesis Test Of Independence*

Description

Perform score and likelihood ratio tests of independence for bivariate data, assuming a logistic dependence model as the alternative.

Usage

```
evind.test(x, method = c("ratio", "score"), verbose = FALSE)
```

Arguments

x	A matrix or data frame, ordinarily with two columns, which may contain missing values.
method	The test methodology; either "ratio" for the likelihood ratio test or "score" for the score test.
verbose	If TRUE, shows estimates of the marginal parameters in addition to the dependence parameter.

Details

This simple function fits a stationary bivariate logistic model to the data and performs a hypothesis test of dep $= 1$ versus dep < 1 using the methodology in Tawn (1988). The null distributions for the printed test statistics are chi-squared on one df for the likelihood ratio test, and standard normal for the score test.

Value

An object of class "htest".

References

Tawn, J. A. (1988) Bivariate extreme value theory: models and estimation. *Biometrika*, **75**, 397–415.

See Also

fbvevd, t.test

Examples

```
evind.test(sealevel)
evind.test(sealevel, method = "score")
```

---

evmc *Simulate Markov Chains With Extreme Value Dependence Structures*

Description

Simulation of first order Markov chains, such that each pair of consecutive values has the dependence structure of one of nine parametric bivariate extreme value distributions.

Usage

```
evmc(n, dep, asy = c(1,1), alpha, beta, model = c("log", "alog",
    "hr", "neglog", "aneglog", "bilog", "negbilog", "ct", "amix"),
    margins = c("uniform","rweibull","frechet","gumbel"))
```

Arguments

n	Number of observations.
dep	Dependence parameter for the logistic, asymmetric logistic, Husler-Reiss, negative logistic and asymmetric negative logistic models.
asy	A vector of length two, containing the two asymmetry parameters for the asymmetric logistic and asymmetric negative logistic models.
alpha, beta	Alpha and beta parameters for the bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models.
model	The specified model; a character string. Must be either "log" (the default), "alog", "hr", "neglog", "aneglog", "bilog", "negbilog", "ct" or "amix" (or any unique partial match), for the logistic, asymmetric logistic, Husler-Reiss, negative logistic, asymmetric negative logistic, bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models respectively. The definition of each model is given in rbvevd. If parameter arguments are given that do not correspond to the specified model those arguments are ignored, with a warning.
margins	The marginal distribution of each value; a character string. Must be either "uniform" (the default), "rweibull", "frechet" or "gumbel" (or any unique partial match), for the uniform, standard reverse Weibull, standard Gumbel and standard Frechet distributions respectively.

Value

A numeric vector of length n.

See Also

marma, rbvevd

Examples

```
evmc(100, alpha = 0.1, beta = 0.1, model = "bilog")
evmc(100, dep = 10, model = "hr", margins = "gum")
```

---

exi *Estimates of the Extremal Index*

Description

Estimates of the extremal index.

Usage

```
exi(data, u, r = 1, ulow = -Inf, rlow = 1)
```

Arguments

data	A numeric vector, which may contain missing values.
u	A single value giving the threshold, unless a time varying threshold is used, in which case u should be a vector of thresholds, typically with the same length as data (or else the usual recycling rules are applied).
r	Either a postive integer denoting the clustering interval length, or zero, in which case the intervals estimator of Ferro and Segers (2003) is used and following arguments are ignored. By default the interval length is one.
ulow	A single value giving the lower threshold, unless a time varying lower threshold is used, in which case ulow should be a vector of lower thresholds, typically with the same length as data (or else the usual recycling rules are applied). By default there is no lower threshold (or equivalently, the lower threshold is -Inf).
rlow	A postive integer denoting the lower clustering interval length. By default the interval length is one.

Details

If r is a positive integer the extremal index is estimated using the inverse of the average cluster size, using the clusters of exceedences derived from clusters. If r is zero, an estimate based on inter-exceedance times is used (Ferro and Segers, 2003).

If there are no exceedances of the threshold, the estimate is NaN. If there is only one exceedance, the estimate is one.

Value

A single value estimating the extremal index.

References

Ferro, C. A. T. and Segers, J. (2003) Inference for clusters of extreme values. *JRSS B*, **65**, 545–556.

See Also

clusters, exiplot

Examples

```
exi(portpirie, 4.2, r = 3, ulow = 3.8)
tvu <- c(rep(4.2, 20), rep(4.1, 25), rep(4.2, 20))
exi(portpirie, tvu, r = 1)
exi(portpirie, tvu, r = 0)
```

---

exiplot *Plot Estimates of the Extremal Index*

Description

Plots estimates of the extremal index.

Usage

```
exiplot(data, tlim, r = 1, ulow = -Inf, rlow = 1, add = FALSE,
    nt = 100, lty = 1, xlab = "Threshold", ylab = "Ext. Index",
    ylim = c(0,1), ...)
```

Arguments

data	A numeric vector, which may contain missing values.
tlim	A numeric vector of length two, giving the limits for the (time invariant) thresholds at which the estimates are evaluated.
r, ulow, rlow	The estimation method. See exi.
add	Add to an existing plot?
nt	The number of thresholds at which the estimates are evaluated.
lty	Line type.
xlab, ylab	x and y axis labels.
ylim	y axis limits.
...	Other arguments passed to plot or lines.

Details

The estimates are calculated using the function exi.

Value

A list with components x and y is invisibly returned. The first component contains the thresholds, the second contains the estimates.

See Also

clusters, exi

Examples

```
sdat <- mar(100, psi = 0.5)
tlim <- quantile(sdat, probs = c(0.4,0.9))
exiplot(sdat, tlim)
exiplot(sdat, tlim, r = 4, add = TRUE, lty = 2)
exiplot(sdat, tlim, r = 0, add = TRUE, lty = 4)
```

---

extreme *Distributions of Maxima and Minima*

Description

Density function, distribution function, quantile function and random generation for the maximum/minimum of a given number of independent variables from a specified distribution.

Usage

```
dextreme(x, densfun, distnfun, ..., distn, mlen = 1, largest = TRUE,
    log = FALSE)
pextreme(q, distnfun, ..., distn, mlen = 1, largest = TRUE,
    lower.tail = TRUE)
qextreme(p, quantfun, ..., distn, mlen = 1, largest = TRUE,
    lower.tail = TRUE)
rextreme(n, quantfun, ..., distn, mlen = 1, largest = TRUE)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
densfun, distnfun, quantfun	Density, distribution and quantile function of the specified distribution. The density function must have a log argument (a simple wrapper can always be constructed to achieve this).

...	Parameters of the specified distribution.
distn	A character string, optionally given as an alternative to densfun, distnfun and quantfun such that the density, distribution and quantile functions are formed upon the addition of the prefixes d, p and q respectively.
mlen	The number of independent variables.
largest	Logical; if TRUE (default) use maxima, otherwise minima.
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default) probabilities are P[X <= x], otherwise P[X > x].

Value

dextreme gives the density function, pextreme gives the distribution function and qextreme gives the quantile function of the maximum/minimum of mlen independent variables from a specified distibution. rextreme generates random deviates.

See Also

rgev, rorder

Examples

```
dextreme(2:4, dnorm, pnorm, mean = 0.5, sd = 1.2, mlen = 5)
dextreme(2:4, distn = "norm", mean = 0.5, sd = 1.2, mlen = 5)
dextreme(2:4, distn = "exp", mlen = 2, largest = FALSE)
pextreme(2:4, distn = "exp", rate = 1.2, mlen = 2)
qextreme(seq(0.9, 0.6, -0.1), distn = "exp", rate = 1.2, mlen = 2)
rextreme(5, qgamma, shape = 1, mlen = 10)
p <- (1:9)/10
pexp(qextreme(p, distn = "exp", rate = 1.2, mlen = 1), rate = 1.2)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

---

failure *Failure Times*

Description

Failure times.

Usage

```
failure
```

Format

A vector containing 24 observations.

Source

van Montfort, M. A. J. and Otten, A. (1978) On testing a shape parameter in the presence of a scale parameter. *Math. Operations Forsch. Statist., Ser. Statistics*, **9**, 91–104.

`fbvevd` *Maximum-likelihood Fitting of Bivariate Extreme Value Distributions*

Description

Fit models for one of nine parametric bivariate extreme value distributions, including linear modelling of the marginal location parameters, and allowing any of the parameters to be held fixed if desired.

Usage

```
fbvevd(x, model = c("log", "alog", "hr", "neglog", "aneglog", "bilog",
    "negbilog", "ct", "amix"), start, ..., sym = FALSE,
    nsloc1 = NULL, nsloc2 = NULL, cshape = cscale, cscale = cloc,
    cloc = FALSE, std.err = TRUE, corr = FALSE, method = "BFGS",
    warn.inf = TRUE)
```

Arguments

`x`	A matrix or data frame, ordinarily with two columns, which may contain missing values. A data frame may also contain a third column of mode `logical`, which itself may contain missing values (see **More Details**).
`model`	The specified model; a character string. Must be either `"log"` (the default), `"alog"`, `"hr"`, `"neglog"`, `"aneglog"`, `"bilog"`, `"negbilog"`, `"ct"` or `"amix"` (or any unique partial match), for the logistic, asymmetric logistic, Husler-Reiss, negative logistic, asymmetric negative logistic, bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models respectively. The definition of each model is given in `rbvevd`.
`start`	A named list giving the initial values for the parameters over which the likelihood is to be maximized. If `start` is omitted the routine attempts to find good starting values using marginal maximum likelihood estimators.
`...`	Additional parameters, either for the bivariate extreme value model or for the optimization function `optim`. If parameters of the model are included they will be held fixed at the values given (see **Examples**).
`sym`	Logical; if `TRUE`, the dependence structure of the models `"alog"`, `"aneglog"` or `"ct"` are constrained to be symmetric (see **Details**). For all other models, the argument is ignored (and a warning is given).
`nsloc1`, `nsloc2`	A data frame with the same number of rows as `x`, for linear modelling of the location parameter on the first/second margin (see **Details**). The data frames are treated as covariate matrices, excluding the intercept. A numeric vector can be given as an alternative to a single column data frame.

cshape	Logical; if TRUE, a common shape parameter is fitted to each margin.
cscale	Logical; if TRUE, a common scale parameter is fitted to each margin, and the default value of cshape is then TRUE, so that under this default common scale and shape parameters are fitted.
cloc	Logical; if TRUE, a common location parameter is fitted to each margin, and the default values of cshape and cscale are then TRUE, so that under these defaults common marginal parameters are fitted.
std.err	Logical; if TRUE (the default), the standard errors are returned.
corr	Logical; if TRUE, the correlation matrix is returned.
method	The optimization method (see optim for details).
warn.inf	Logical; if TRUE (the default), a warning is given if the negative log-likelihood is infinite when evaluated at the starting values.

Details

The dependence parameter names are one or more of dep, asy1, asy2, alpha and beta, depending on the model selected (see rbvevd). The marginal parameter names are loc1, scale1 and shape1 for the first margin, and loc2, scale2 and shape2 for the second margin. If nsloc1 is not NULL, so that a linear model is implemented for the first marginal location parameter, the parameter names for the first margin are loc1, loc1*x1*, ..., loc1*xn*, scale and shape, where *x1*, ..., *xn* are the column names of nsloc1, so that loc1 is the intercept of the linear model, and loc1*x1*, ..., loc1*xn* are the ncol(nsloc1) coefficients. When nsloc2 is not NULL, the parameter names for the second margin are constructed similarly.

It is recommended that the covariates within the linear models for the location parameters are (at least approximately) centered and scaled (i.e. that the columns of nsloc1 and nsloc2 are centered and scaled), particularly if automatic starting values are used, since the starting values for the associated parameters are then zero. If cloc is TRUE, both nsloc1 and nsloc2 must be identical, since a common linear model is then implemented on both margins.

If cshape is true, the models are constrained so that shape2 = shape1. The parameter shape2 is then taken to be specified, so that e.g. the common shape parameter can only be fixed at zero using shape1 = 0, since using shape2 = 0 gives an error. Similar comments apply for cscale and cloc.

If sym is TRUE, the asymmetric logistic and asymmetric negative logistic models are constrained so that asy2 = asy1, and the Coles-Tawn model is constrained so that beta = alpha. The parameter asy2 or beta is then taken to be specified, so that e.g. the parameters asy1 and asy2 can only be fixed at 0.8 using asy1 = 0.8, since using asy2 = 0.8 gives an error.

Bilogistic and negative bilogistic models constrained to symmetry are logistic and negative logistic models respectively. The (symmetric) mixed model (e.g. Tawn, 1998) can be obtained as a special case of the asymmetric logistic or asymmetric mixed models (see **Examples**).

The value Dependence given in the printed output is $2(1 - A(1/2))$, where A is the estimated dependence function (see abvevd). It measures the strength of dependence, and lies in the interval [0,1]; at independence and complete dependence it is zero and one respectively (Coles, Heffernan and Tawn, 1999). See chiplot for further information.

Value

Returns an object of class `c("bvevd","evd")`.

The generic accessor functions `fitted` (or `fitted.values`), `std.errors`, `deviance`, `logLik` and `AIC` extract various features of the returned object.

The functions `profile` and `profile2d` can be used to obtain deviance profiles. The function `anova` compares nested models, and the function `AIC` compares non-nested models. The function `plot` produces diagnostic plots.

An object of class `c("bvevd","evd")` is a list containing the following components

`estimate`	A vector containing the maximum likelihood estimates.
`std.err`	A vector containing the standard errors.
`fixed`	A vector containing the parameters that have been fixed at specific values within the optimization.
`fixed2`	A vector containing the parameters that have been set to be equal to other model parameters.
`param`	A vector containing all parameters (those optimized, those fixed to specific values, and those set to be equal to other model parameters).
`deviance`	The deviance at the maximum likelihood estimates.
`dep.summary`	The estimate of $2(1 - A(1/2))$.
`corr`	The correlation matrix.
`var.cov`	The variance covariance matrix.
`convergence`, `counts`, `message`	Components taken from the list returned by `optim`.
`data`	The data passed to the argument `x`.
`tdata`	The data, transformed to stationarity (for non-stationary models).
`nsloc1`, `nsloc2`	The arguments `nsloc1` and `nsloc2`.
`n`	The number of rows in `x`.
`sym`	The argument `sym`.
`cmar`	The vector `c(cloc, cscale, cshape)`.
`model`	The argument `model`.
`call`	The call of the current function.

More Details

If `x` is a data frame with a third column of mode `logical`, then the model is fitted using the likelihood derived by Stephenson and Tawn (2004). This is appropriate when each bivariate data point comprises componentwise maxima from some underlying bivariate process, and where the corresponding logical value denotes whether or not the maxima were caused by the same event within that process.

Under this scheme the diagnostic plots that are produced using `plot` are somewhat different to those described in `plot.bvevd`: the density, dependence function and quantile curves plots contain fitted functions for observations where the logical case is unknown, and the conditional P-P plots condition on both the logical case and the given margin (which requires numerical integration at each data point).

Artificial Constraints

For numerical reasons parameters are subject to artificial constraints. Specifically, these constraints are: marginal scale parameters not less than 0.01; dep not less than [0.1] [0.2] [0.05] in [logistic] [Husler-Reiss] [negative logistic] models; dep not greater than [10] [5] in [Husler-Reiss] [negative logistic] models; asy1 and asy2 not less than 0.001; alpha and beta not less than [0.1] [0.1] [0.001] in [bilogistic] [negative bilogistic] [Coles-Tawn] models; alpha and beta not greater than [0.999] [20] [30] in [bilogistic] [negative bilogistic] [Coles-Tawn] models.

Warning

The standard errors and the correlation matrix in the returned object are taken from the observed information, calculated by a numerical approximation. They must be interpreted with caution when either of the marginal shape parameters are less than -0.5, because the usual asymptotic properties of maximum likelihood estimators do not then hold (Smith, 1985).

References

Coles, S. G., Heffernan, J. and Tawn, J. A. (1999) Dependence measures for extreme value analyses. *Extremes*, **2**, 339–365.

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

Stephenson, A. G. and Tawn, J. A. (2004) Exploiting Occurence Times in Likelihood Inference for Componentwise Maxima. *Biometrika* **92**(1), 213–217.

Tawn, J. A. (1988) Bivariate extreme value theory: models and estimation. *Biometrika*, **75**, 397–415.

See Also

anova.evd, optim, plot.bvevd, profile.evd, profile2d.evd, rbvevd

Examples

```
bvdata <- rbvevd(100, dep = 0.6, model = "log", mar1 = c(1.2,1.4,0.4))
M1 <- fbvevd(bvdata, model = "log")
M2 <- fbvevd(bvdata, model = "log", dep = 0.75)
anova(M1, M2)
par(mfrow = c(2,2))
plot(M1)
plot(M1, mar = 1)
plot(M1, mar = 2)
## Not run: par(mfrow = c(1,1))
## Not run: M1P <- profile(M1, which = "dep")
## Not run: plot(M1P)

trend <- (-49:50)/100
rnd <- runif(100, min = -.5, max = .5)
fbvevd(bvdata, model = "log", nsloc1 = trend)
fbvevd(bvdata, model = "log", nsloc1 = trend, nsloc2 = data.frame(trend
= trend,  random = rnd))
```

```
fbvevd(bvdata, model = "log", nsloc1 = trend, nsloc2 = data.frame(trend
= trend, random = rnd), loc2random = 0)

bvdata <- rbvevd(100, dep = 1, asy = c(0.5,0.5), model = "anegl")
anlog <- fbvevd(bvdata, model = "anegl")
mixed <- fbvevd(bvdata, model = "anegl", dep = 1, sym = TRUE)
anova(anlog, mixed)
amixed <- fbvevd(bvdata, model = "amix")
mixed <- fbvevd(bvdata, model = "amix", beta = 0)
anova(amixed, mixed)
```

`fbvpot` *Maximum-likelihood Fitting of Bivariate Extreme Value Distributions to Threshold Exceedances*

Description

Fit models for one of nine parametric bivariate extreme-value distributions using threshold exceedances, allowing any of the parameters to be held fixed if desired.

Usage

```
fbvpot(x, threshold, model = c("log", "bilog", "alog", "neglog",
    "negbilog", "aneglog", "ct", "hr", "amix"), likelihood =
    c("censored", "poisson"), start, ..., sym = FALSE, cshape =
    cscale, cscale = FALSE, std.err = TRUE, corr = FALSE, method =
    "BFGS", warn.inf = TRUE)
```

Arguments

`x`	A matrix or data frame with two columns. If this contains missing values, those values are treated as if they fell below the corresponding marginal threshold.
`threshold`	A vector of two thresholds.
`model`	The specified model; a character string. Must be either `"log"` (the default), `"alog"`, `"hr"`, `"neglog"`, `"aneglog"`, `"bilog"`, `"negbilog"`, `"ct"` or `"amix"` (or any unique partial match), for the logistic, asymmetric logistic, Husler-Reiss, negative logistic, asymmetric negative logistic, bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models respectively. The definition of each model is given in `rbvevd`.
`likelihood`	The likelihood model; either `"censored"` (the default) or `"poisson"`. The `"poisson"` method is not recommended. See **Details**.
`start`	A named list giving the initial values for all of the parameters in the model. If `start` is omitted the routine attempts to find good starting values using marginal maximum likelihood estimators.
`...`	Additional parameters, either for the bivariate extreme value model or for the optimization function `optim`. If parameters of the model are included they will be held fixed at the values given (see **Examples**).

sym	Logical; if TRUE, the dependence structure of the models "alog", "aneglog" or "ct" are constrained to be symmetric (see **Details**). For all other models, the argument is ignored (and a warning is given).
cshape	Logical; if TRUE, a common shape parameter is fitted to each margin.
cscale	Logical; if TRUE, a common scale parameter is fitted to each margin, and the default value of cshape is then TRUE, so that under this default common marginal parameters are fitted.
std.err	Logical; if TRUE (the default), the standard errors are returned.
corr	Logical; if TRUE, the correlation matrix is returned.
method	The optimization method (see optim for details).
warn.inf	Logical; if TRUE (the default), a warning is given if the negative log-likelihood is infinite when evaluated at the starting values.

Details

For the "censored" method bivariate peaks over threshold models are fitted by maximizing the censored likelihood as given in e.g. Section 8.3.1 of Coles(2001). For the "poisson" method models are fitted using Equation 5.4 of Coles and Tawn (1991), see also Joe, Smith and Weissman (1992). This method is only available for models whose spectral measure does not contain point masses (see hbvevd). It is not recommended as in practice it can produce poor estimates.

For either likelihood the margins are modelled using a generalized Pareto distribution for points above the threshold and an empirical model for those below. For the "poisson" method data lying below both thresholds is not used. For the "censored" method the number of points lying below both thresholds is used, but the locations of the those points are not.

The dependence parameter names are one or more of dep, asy1, asy2, alpha and beta, depending on the model selected (see rbvevd). The marginal parameter names are scale1 and shape1 for the first margin, and scale2 and shape2 for the second margin.

If cshape is true, the models are constrained so that shape2 = shape1. The parameter shape2 is then taken to be specified, so that e.g. the common shape parameter can only be fixed at zero using shape1 = 0, since using shape2 = 0 gives an error. Similar comments apply for cscale.

If sym is TRUE, the asymmetric logistic and asymmetric negative logistic models are constrained so that asy2 = asy1, and the Coles-Tawn model is constrained so that beta = alpha. The parameter asy2 or beta is then taken to be specified, so that e.g. the parameters asy1 and asy2 can only be fixed at 0.8 using asy1 = 0.8, since using asy2 = 0.8 gives an error.

Bilogistic and negative bilogistic models constrained to symmetry are logistic and negative logistic models respectively. The (symmetric) mixed model (e.g. Tawn, 1998) can be obtained as a special case of the asymmetric logistic or asymmetric mixed models (see **fbvevd**).

For numerical reasons the parameters of each model are subject the artificial constraints given in fbvevd.

Value

Returns an object of class c("bvpot","evd").

The generic accessor functions fitted (or fitted.values), std.errors, deviance, logLik and AIC extract various features of the returned object.

The functions `profile` and `profile2d` can be used to obtain deviance profiles. The function `anova` compares nested models, and the function `AIC` compares non-nested models. There is currently no plot method available.

An object of class `c("bvpot","evd")` is a list containing the following components

`estimate`	A vector containing the maximum likelihood estimates.
`std.err`	A vector containing the standard errors.
`fixed`	A vector containing the parameters that have been fixed at specific values within the optimization.
`fixed2`	A vector containing the parameters that have been set to be equal to other model parameters.
`param`	A vector containing all parameters (those optimized, those fixed to specific values, and those set to be equal to other model parameters).
`deviance`	The deviance at the maximum likelihood estimates.
`dep.summary`	A value summarizing the strength of dependence in the fitted model (see **fbvevd**).
`corr`	The correlation matrix.
`var.cov`	The variance covariance matrix.
`convergence`, `counts`, `message`	Components taken from the list returned by `optim`.
`data`	The data passed to the argument `x`.
`threshold`	The argument `threshold`.
`n`	The number of rows in `x`.
`nat`	The vector of length three containing the number of exceedances on the first, second and both margins respectively.
`likelihood`	The argument `likelihood`.
`sym`	The argument `sym`.
`cmar`	The vector `c(cscale, cshape)`.
`model`	The argument `model`.
`call`	The call of the current function.

Warning

The standard errors and the correlation matrix in the returned object are taken from the observed information, calculated by a numerical approximation. They must be interpreted with caution when either of the marginal shape parameters are less than -0.5, because the usual asymptotic properties of maximum likelihood estimators do not then hold (Smith, 1985).

Author(s)

Chris Ferro and Alec Stephenson

References

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values*, London: Springer–Verlag.

Coles, S. G. and Tawn, J. A. (1991) Modelling multivariate extreme events. *J. R. Statist. Soc. B*, **53**, 377–392.

Joe, H., Smith, R. L. and Weissman, I. (1992) Bivariate threshold methods for extremes. *J. R. Statist. Soc. B*, **54**, 171–183.

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

See Also

abvevd, anova.evd, fbvevd, optim, rbvevd

Examples

```
bvdata <- rbvevd(1000, dep = 0.5, model = "log")
u <- apply(bvdata, 2,  quantile, probs = 0.9)
M1 <- fbvpot(bvdata, u, model = "log")
M2 <- fbvpot(bvdata, u, "log", dep = 0.5)
anova(M1, M2)
```

---

fextreme *Maximum-likelihood Fitting of Maxima and Minima*

Description

Maximum-likelihood fitting for the distribution of the maximum/minimum of a given number of independent variables from a specified distribution.

Usage

```
fextreme(x, start, densfun, distnfun, ..., distn, mlen = 1, largest =
    TRUE, std.err = TRUE, corr = FALSE, method = "Nelder-Mead")
```

Arguments

x	A numeric vector.
start	A named list giving the initial values for the parameters over which the likelihood is to be maximized.
densfun, distnfun	Density and distribution function of the specified distribution.

...	Additional parameters, either for the specified distribution or for the optimization function optim. If parameters of the distribution are included they will be held fixed at the values given (see **Examples**). If parameters of the distribution are not included either here or as a named component in start they will be held fixed at the default values specified in the corresponding density and distribution functions (assuming they exist; an error will be generated otherwise).
distn	A character string, optionally specified as an alternative to densfun and distnfun such that the density and distribution functions are formed upon the addition of the prefixes d and p respectively.
mlen	The number of independent variables.
largest	Logical; if TRUE (default) use maxima, otherwise minima.
std.err	Logical; if TRUE (the default), the standard errors are returned.
corr	Logical; if TRUE, the correlation matrix is returned.
method	The optimization method (see optim for details).

Details

Maximization of the log-likelihood is performed. The estimated standard errors are taken from the observed information, calculated by a numerical approximation.

If the density and distribution functions are user defined, the order of the arguments must mimic those in R base (i.e. data first, parameters second). Density functions must have log arguments.

Value

Returns an object of class c("extreme","evd").

The generic accessor functions fitted (or fitted.values), std.errors, deviance, logLik and AIC extract various features of the returned object. The function anova compares nested models.

An object of class c("extreme","evd") is a list containing at most the following components

estimate	A vector containing the maximum likelihood estimates.
std.err	A vector containing the standard errors.
deviance	The deviance at the maximum likelihood estimates.
corr	The correlation matrix.
var.cov	The variance covariance matrix.
convergence, counts, message	Components taken from the list returned by optim.
call	The call of the current function.
data	The data passed to the argument x.
n	The length of x.

See Also

anova.evd, forder, optim

Examples

```
uvdata <- rextreme(100, qnorm, mean = 0.56, mlen = 365)
fextreme(uvdata, list(mean = 0, sd = 1), distn = "norm", mlen = 365)
fextreme(uvdata, list(rate = 1), distn = "exp", mlen = 365,
  method = "Brent", lower=0.01, upper=10)
fextreme(uvdata, list(scale = 1), shape = 1, distn = "gamma", mlen = 365,
  method = "Brent", lower=0.01, upper=10)
fextreme(uvdata, list(shape = 1, scale = 1), distn = "gamma", mlen = 365)
```

`fgev` *Maximum-likelihood Fitting of the Generalized Extreme Value Distribution*

Description

Maximum-likelihood fitting for the generalized extreme value distribution, including linear modelling of the location parameter, and allowing any of the parameters to be held fixed if desired.

Usage

```
fgev(x, start, ..., nsloc = NULL, prob = NULL, std.err = TRUE,
    corr = FALSE, method = "BFGS", warn.inf = TRUE)
```

Arguments

`x`	A numeric vector, which may contain missing values.
`start`	A named list giving the initial values for the parameters over which the likelihood is to be maximized. If `start` is omitted the routine attempts to find good starting values using moment estimators.
`...`	Additional parameters, either for the GEV model or for the optimization function `optim`. If parameters of the model are included they will be held fixed at the values given (see **Examples**).
`nsloc`	A data frame with the same number of rows as the length of `x`, for linear modelling of the location parameter. The data frame is treated as a covariate matrix (excluding the intercept). A numeric vector can be given as an alternative to a single column data frame.
`prob`	Controls the parameterization of the model (see **Details**). Should be either `NULL` (the default), or a probability in the closed interval [0,1].
`std.err`	Logical; if `TRUE` (the default), the standard errors are returned.
`corr`	Logical; if `TRUE`, the correlation matrix is returned.
`method`	The optimization method (see `optim` for details).
`warn.inf`	Logical; if `TRUE` (the default), a warning is given if the negative log-likelihood is infinite when evaluated at the starting values.

Details

If prob is NULL (the default):

For stationary models the parameter names are loc, scale and shape, for the location, scale and shape parameters respectively. For non-stationary models, the parameter names are loc, loc*x1*, ..., loc*xn*, scale and shape, where *x1*, ..., *xn* are the column names of nsloc, so that loc is the intercept of the linear model, and loc*x1*, ..., loc*xn* are the ncol(nsloc) coefficients. If nsloc is a vector it is converted into a single column data frame with column name trend, and hence the associated trend parameter is named loctrend.

If prob $= p$ is a probability:

The fit is performed using a different parameterization. Let a, b and s denote the location, scale and shape parameters of the GEV distribution. For stationary models, the distribution is parameterized using (z_p, b, s), where

$$z_p = a - b/s(1 - (-\log(1-p))^s)$$

is such that $G(z_p) = 1 - p$, where G is the GEV distribution function. prob $= p$ is therefore the probability in the upper tail corresponding to the quantile z_p. If prob is zero, then z_p is the upper end point $a - b/s$, and s is restricted to the negative (Weibull) axis. If prob is one, then z_p is the lower end point $a - b/s$, and s is restricted to the positive (Frechet) axis. The parameter names are quantile, scale and shape, for z_p, b and s respectively.

For non-stationary models the parameter z_p is again given by the equation above, but a becomes the intercept of the linear model for the location parameter, so that quantile replaces (the intercept) loc, and hence the parameter names are quantile, loc*x1*, ..., loc*xn*, scale and shape, where *x1*, ..., *xn* are the column names of nsloc.

In either case:

For non-stationary fitting it is recommended that the covariates within the linear model for the location parameter are (at least approximately) centered and scaled (i.e.\ that the columns of nsloc are centered and scaled), particularly if automatic starting values are used, since the starting values for the associated parameters are then zero.

Value

Returns an object of class c("gev","uvevd","evd").

The generic accessor functions fitted (or fitted.values), std.errors, deviance, logLik and AIC extract various features of the returned object.

The functions profile and profile2d are used to obtain deviance profiles for the model parameters. In particular, profiles of the quantile z_p can be calculated and plotted when prob $= p$. The function anova compares nested models. The function plot produces diagnostic plots.

An object of class c("gev","uvevd","evd") is a list containing at most the following components

estimate	A vector containing the maximum likelihood estimates.
std.err	A vector containing the standard errors.
fixed	A vector containing the parameters of the model that have been held fixed.
param	A vector containing all parameters (optimized and fixed).
deviance	The deviance at the maximum likelihood estimates.
corr	The correlation matrix.

`var.cov` The variance covariance matrix.

`convergence`, `counts`, `message`
Components taken from the list returned by `optim`.

`data` The data passed to the argument `x`.

`tdata` The data, transformed to stationarity (for non-stationary models).

`nsloc` The argument `nsloc`.

`n` The length of `x`.

`prob` The argument `prob`.

`loc` The location parameter. If `prob` is `NULL` (the default), this will also be an element of `param`.

`call` The call of the current function.

Warning

The standard errors and the correlation matrix in the returned object are taken from the observed information, calculated by a numerical approximation. They must be interpreted with caution when the shape parameter is less than -0.5, because the usual asymptotic properties of maximum likelihood estimators do not then hold (Smith, 1985).

References

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

See Also

`anova.evd`, `optim`, `plot.uvevd`, `profile.evd`, `profile2d.evd`

Examples

```
uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
trend <- (-49:50)/100
M1 <- fgev(uvdata, nsloc = trend, control = list(trace = 1))
M2 <- fgev(uvdata)
M3 <- fgev(uvdata, shape = 0)
M4 <- fgev(uvdata, scale = 1, shape = 0)
anova(M1, M2, M3, M4)
par(mfrow = c(2,2))
plot(M2)
## Not run: M2P <- profile(M2)
## Not run: plot(M2P)

rnd <- runif(100, min = -.5, max = .5)
fgev(uvdata, nsloc = data.frame(trend = trend, random = rnd))
fgev(uvdata, nsloc = data.frame(trend = trend, random = rnd), locrandom = 0)

uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
M1 <- fgev(uvdata, prob = 0.1)
M2 <- fgev(uvdata, prob = 0.01)
```

```
## Not run: M1P <- profile(M1, which = "quantile")
## Not run: M2P <- profile(M2, which = "quantile")
## Not run: plot(M1P)
## Not run: plot(M2P)
```

`fgumbelx` *Maximum-likelihood Fitting of the Maximum of Two Gumbel Distributions*

Description

Maximum-likelihood fitting for the maximum of two gumbel distributions, allowing any of the parameters to be held fixed if desired.

Usage

```
fgumbelx(x, start, ..., nsloc1 = NULL, nsloc2 = NULL, std.err = TRUE,
    corr = FALSE, method = "BFGS", warn.inf = TRUE)
```

Arguments

`x`	A numeric vector, which may contain missing values.
`start`	A named list giving the initial values for the parameters over which the likelihood is to be maximized. If `start` is omitted the routine attempts to find good starting values using moment estimators.
`...`	Additional parameters, either for the fitted model or for the optimization function `optim`. If parameters of the model are included they will be held fixed at the values given (see **Examples**).
`nsloc1`	A data frame with the same number of rows as the length of `x`, for linear modelling of the location parameter of the first Gumbel distribution. This is not recommended as the model is already complex.
`nsloc2`	A data frame with the same number of rows as the length of `x`, for linear modelling of the location parameter of the second Gumbel distribution. This is not recommended as the model is already complex.
`std.err`	Logical; if `TRUE` (the default), the standard errors are returned.
`corr`	Logical; if `TRUE`, the correlation matrix is returned.
`method`	The optimization method (see `optim` for details).
`warn.inf`	Logical; if `TRUE` (the default), a warning is given if the negative log-likelihood is infinite when evaluated at the starting values.

Details

For stationary models the parameter names are loc1, scale1, loc2 and scale2 for the location and scale parameters of two Gumbel distributions, where loc2 must be greater or equal to loc1.

The likelihood may have multiple local optima and therefore may be difficult to fit properly; the default starting values use a moment based approach, however it is recommended that the user specify multiple different starting values and experiment with different optimization methods.

Using non-stationary models with nsloc1 and nsloc2 is not recommended due to the model complexity; the data also cannot be transformed back to stationarity so diagnostic plots will be misleading in this case.

Value

Returns an object of class c("gumbelx","evd").

The generic accessor functions fitted (or fitted.values), std.errors, deviance, logLik and AIC extract various features of the returned object.

The functions profile and profile2d are used to obtain deviance profiles for the model parameters. The function anova compares nested models. The function plot produces diagnostic plots.

An object of class c("gumbelx","evd") is a list containing at most the following components

estimate	A vector containing the maximum likelihood estimates.
std.err	A vector containing the standard errors.
fixed	A vector containing the parameters of the model that have been held fixed.
param	A vector containing all parameters (optimized and fixed).
deviance	The deviance at the maximum likelihood estimates.
corr	The correlation matrix.
var.cov	The variance covariance matrix.
convergence, counts, message	Components taken from the list returned by optim.
data	The data passed to the argument x.
nsloc1	The argument nsloc1.
nsloc2	The argument nsloc2.
n	The length of x.
call	The call of the current function.

Warning

This function is experimental and involves optimizing over a potentially complex surface.

See Also

fgev, optim, rgumbelx

Examples

```
uvdata <- rgumbelx(100, loc1 = 0, scale1 = 1, loc2 = 1, scale2 = 1)
fgumbelx(uvdata, loc1 = 0, scale1 = 1)
```

forder *Maximum-likelihood Fitting of Order Statistics*

Description

Maximum-likelihood fitting for the distribution of a selected order statistic of a given number of independent variables from a specified distribution.

Usage

```
forder(x, start, densfun, distnfun, ..., distn, mlen = 1, j = 1,
    largest = TRUE, std.err = TRUE, corr = FALSE, method = "Nelder-Mead")
```

Arguments

x	A numeric vector.
start	A named list giving the initial values for the parameters over which the likelihood is to be maximized.
densfun, distnfun	Density and distribution function of the specified distribution.
...	Additional parameters, either for the specified distribution or for the optimization function optim. If parameters of the distribution are included they will be held fixed at the values given (see **Examples**). If parameters of the distribution are not included either here or as a named component in start they will be held fixed at the default values specified in the corresponding density and distribution functions (assuming they exist; an error will be generated otherwise).
distn	A character string, optionally specified as an alternative to densfun and distnfun such that the density and distribution and functions are formed upon the addition of the prefixes d and p respectively.
mlen	The number of independent variables.
j	The order statistic, taken as the jth largest (default) or smallest of mlen, according to the value of largest.
largest	Logical; if TRUE (default) use the jth largest order statistic, otherwise use the jth smallest.
std.err	Logical; if TRUE (the default), the standard errors are returned.
corr	Logical; if TRUE, the correlation matrix is returned.
method	The optimization method (see optim for details).

Details

Maximization of the log-likelihood is performed. The estimated standard errors are taken from the observed information, calculated by a numerical approximation.

If the density and distribution functions are user defined, the order of the arguments must mimic those in R base (i.e. data first, parameters second). Density functions must have log arguments.

Value

Returns an object of class `c("extreme","evd")`. This class is defined in `fextreme`.

The generic accessor functions `fitted` (or `fitted.values`), `std.errors`, `deviance`, `logLik` and `AIC` extract various features of the returned object. The function `anova` compares nested models.

See Also

`anova.evd`, `fextreme`, `optim`

Examples

```
uvd <- rorder(100, qnorm, mean = 0.56, mlen = 365, j = 2)
forder(uvd, list(mean = 0, sd = 1), distn = "norm", mlen = 365, j = 2)
forder(uvd, list(rate = 1), distn = "exp", mlen = 365, j = 2,
  method = "Brent", lower=0.01, upper=10)
forder(uvd, list(scale = 1), shape = 1, distn = "gamma", mlen = 365, j = 2,
  method = "Brent", lower=0.01, upper=10)
forder(uvd, list(shape = 1, scale = 1), distn = "gamma", mlen = 365, j = 2)
```

---

fox *Maximum Annual Flood Discharges of the Fox River*

Description

The `fox` data frame has 33 rows and 2 columns. The columns contain maximum annual flood discharges, in units of 1000 cubed feet per second, from the Fox River in Wisconsin, USA at Berlin (upstream) and Wrightstown (downstream), for the years 1918 to 1950. The row names give the years of observation.

Usage

```
fox
```

Format

This data frame contains the following columns:

berlin A numeric vector containing maximum annual flood discharges at Berlin (upstream).

wright A numeric vector containing maximum annual flood discharges at Wrightstown (downstream).

Source

Gumbel, E. J. and Mustafi, C. K. (1967) Some analytical properties of bivariate extremal distributions. *J. Amer. Statist. Assoc.*, **62**, 569–588.

fpot *Peaks Over Threshold Modelling using the Generalized Pareto or Point Process Representation*

Description

Maximum-likelihood fitting for peaks over threshold modelling, using the Generalized Pareto or Point Process representation, allowing any of the parameters to be held fixed if desired.

Usage

```
fpot(x, threshold, model = c("gpd", "pp"), start, npp = length(x),
    cmax = FALSE, r = 1, ulow = -Inf, rlow = 1, mper = NULL, ...,
    std.err = TRUE, corr = FALSE, method = "BFGS", warn.inf = TRUE)
```

Arguments

x	A numeric vector. If this contains missing values, those values are treated as if they fell below the threshold.
threshold	The threshold.
model	The model; either "gpd" (the default) or "pp", for the Generalized Pareto or Point Process representations respectively.
start	A named list giving the initial values for the parameters over which the likelihood is to be maximized. If start is omitted the routine attempts to find good starting values using moment estimators.
npp	The data should contain npp observations per "period", where the return level plot produced by plot.pot will represent return periods in units of "periods". By default npp = length(x), so that the "period" is the period of time over which the entire data set is collected. It may often be useful to change this default so that more sensible units are used. For example, if yearly periodic units are required, use npp = 365.25 for daily data and npp = 52.18 for weekly data. The argument only makes a difference to the actual fit if mper is not NULL or if model = "pp" (see **Details**).
cmax	Logical; if FALSE (the default), the model is fitted using all exceedences over the threshold. If TRUE, the model is fitted using cluster maxima, using clusters of exceedences derived from clusters.
r, ulow, rlow	Arguments used for the identification of clusters of exceedences (see clusters). Ignored if cmax is FALSE (the default).
mper	Controls the parameterization of the generalized Pareto model. Should be either NULL (the default), or a positive number (see **Details**). If mper is not NULL and model = "pp", an error is returned.
...	Additional parameters, either for the model or for the optimization function optim. If parameters of the model are included they will be held fixed at the values given (see **Examples**).

std.err	Logical; if TRUE (the default), the standard errors are returned.
corr	Logical; if TRUE, the correlation matrix is returned.
method	The optimization method (see optim for details).
warn.inf	Logical; if TRUE (the default), a warning is given if the negative log-likelihood is infinite when evaluated at the starting values.

Details

The exeedances over the threshold threshold (if cmax is FALSE) or the maxima of the clusters of exeedances (if cmax is TRUE) are (if model = "gpd") fitted to a generalized Pareto distribution (GPD) with location threshold. If model = "pp" the exceedances are fitted to a non-homogeneous Poisson process (Coles, 2001).

If mper is NULL (the default), the parameters of the model (if model = "gpd") are scale and shape, for the scale and shape parameters of the GPD. If model = "pp" the parameters are loc, scale and shape. Under model = "pp" the parameters can be interpreted as parameters of the Generalized Extreme Value distribution, fitted to the maxima of npp random variables. In this case, the value of npp should be reasonably large.

For both characterizations, the shape parameters are equivalent. The scale parameter under the generalized Pareto characterization is equal to $b + s(u - a)$, where a, b and s are the location, scale and shape parameters under the Point Process characterization, and where u is the threshold.

If mper $= m$ is a positive value, then the generalized Pareto model is reparameterized so that the parameters are rlevel and shape, where rlevel is the m "period" return level, where "period" is defined via the argument npp.

The m "period" return level is defined as follows. Let G be the fitted generalized Pareto distribution function, with location threshold $= u$, so that $1 - G(z)$ is the fitted probability of an exceedance over $z > u$ given an exceedance over u. The fitted probability of an exceedance over $z > u$ is therefore $p(1 - G(z))$, where p is the estimated probabilty of exceeding u, which is given by the empirical proportion of exceedances. The m "period" return level z_m satisfies $p(1 - G(z_m)) = 1/(mN)$, where N is the number of points per period (multiplied by the estimate of the extremal index, if cluster maxima are fitted). In other words, z_m is the quantile of the fitted model that corresponds to the upper tail probability $1/(mN)$. If mper is infinite, then z_m is the upper end point, given by threshold minus scale/shape, and the shape parameter is then restricted to be negative.

Value

Returns an object of class c("pot","uvevd","pot").

The generic accessor functions fitted (or fitted.values), std.errors, deviance, logLik and AIC extract various features of the returned object.

The function profile can be used to obtain deviance profiles for the model parameters. In particular, profiles of the m period return level z_m can be calculated and plotted when mper $= m$. The function anova compares nested models. The function plot produces diagnostic plots.

An object of class c("pot","uvevd","evd") is a list containing the following components

estimate	A vector containing the maximum likelihood estimates.
std.err	A vector containing the standard errors.

`fixed`	A vector containing the parameters of the model that have been held fixed.
`param`	A vector containing all parameters (optimized and fixed).
`deviance`	The deviance at the maximum likelihood estimates.
`corr`	The correlation matrix.
`var.cov`	The variance covariance matrix.
`convergence`, `counts`, `message`	Components taken from the list returned by `optim`.
`threshold`, `r`, `ulow`, `rlow`, `npp`	The arguments of the same name.
`nhigh`	The number of exceedences (if `cmax` is `FALSE`) or the number of clusters of exceedences (if `cmax` is `TRUE`).
`nat`, `pat`	The number and proportion of exceedences.
`extind`	The estimate of the extremal index (i.e. `nhigh` divided by `nat`). If `cmax` is `FALSE`, this is `NULL`.
`data`	The data passed to the argument `x`.
`exceedances`	The exceedences, or the maxima of the clusters of exceedences.
`mper`	The argument `mper`.
`scale`	The scale parameter for the fitted generalized Pareto distribution. If `mper` is `NULL` and `model = "gpd"` (the defaults), this will also be an element of `param`.
`call`	The call of the current function.

Warning

The standard errors and the correlation matrix in the returned object are taken from the observed information, calculated by a numerical approximation. They must be interpreted with caution when the shape parameter is less than -0.5, because the usual asymptotic properties of maximum likelihood estimators do not then hold (Smith, 1985).

References

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

See Also

`anova.evd`, `optim`, `plot.uvevd`, `profile.evd`, `profile2d.evd`, `mrlplot`, `tcplot`

Examples

```
uvdata <- rgpd(100, loc = 0, scale = 1.1, shape = 0.2)
M1 <- fpot(uvdata, 1)
M2 <- fpot(uvdata, 1, shape = 0)
anova(M1, M2)
par(mfrow = c(2,2))
plot(M1)
## Not run: M1P <- profile(M1)
```

```
## Not run: plot(M1P)

M1 <- fpot(uvdata, 1, mper = 10)
M2 <- fpot(uvdata, 1, mper = 100)
## Not run: M1P <- profile(M1, which = "rlevel", conf=0.975, mesh=0.1)
## Not run: M2P <- profile(M2, which = "rlevel", conf=0.975, mesh=0.1)
## Not run: plot(M1P)
## Not run: plot(M2P)
```

`frechet` *The Frechet Distribution*

Description

Density function, distribution function, quantile function and random generation for the Frechet distribution with location, scale and shape parameters.

Usage

```
dfrechet(x, loc=0, scale=1, shape=1, log = FALSE)
pfrechet(q, loc=0, scale=1, shape=1, lower.tail = TRUE)
qfrechet(p, loc=0, scale=1, shape=1, lower.tail = TRUE)
rfrechet(n, loc=0, scale=1, shape=1)
```

Arguments

`x`, `q`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations.
`loc`, `scale`, `shape`	Location, scale and shape parameters (can be given as vectors).
`log`	Logical; if `TRUE`, the log density is returned.
`lower.tail`	Logical; if `TRUE` (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The Frechet distribution function with parameters `loc` $= a$, `scale` $= b$ and `shape` $= s$ is

$$G(z) = \exp\left\{-\left(\frac{z-a}{b}\right)^{-s}\right\}$$

for $z > a$ and zero otherwise, where $b > 0$ and $s > 0$.

Value

`dfrechet` gives the density function, `pfrechet` gives the distribution function, `qfrechet` gives the quantile function, and `rfrechet` generates random deviates.

See Also

rgev, rgumbel, rrweibull

Examples

```
dfrechet(2:4, 1, 0.5, 0.8)
pfrechet(2:4, 1, 0.5, 0.8)
qfrechet(seq(0.9, 0.6, -0.1), 2, 0.5, 0.8)
rfrechet(6, 1, 0.5, 0.8)
p <- (1:9)/10
pfrechet(qfrechet(p, 1, 2, 0.8), 1, 2, 0.8)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

---

gev *The Generalized Extreme Value Distribution*

Description

Density function, distribution function, quantile function and random generation for the generalized extreme value (GEV) distribution with location, scale and shape parameters.

Usage

```
dgev(x, loc=0, scale=1, shape=0, log = FALSE)
pgev(q, loc=0, scale=1, shape=0, lower.tail = TRUE)
qgev(p, loc=0, scale=1, shape=0, lower.tail = TRUE)
rgev(n, loc=0, scale=1, shape=0)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
loc, scale, shape	Location, scale and shape parameters; the shape argument cannot be a vector (must have length one).
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The GEV distribution function with parameters loc $= a$, scale $= b$ and shape $= s$ is

$$G(z) = \exp\left[-\{1 + s(z-a)/b\}^{-1/s}\right]$$

for $1 + s(z - a)/b > 0$, where $b > 0$. If $s = 0$ the distribution is defined by continuity. If $1 + s(z - a)/b \leq 0$, the value z is either greater than the upper end point (if $s < 0$), or less than the lower end point (if $s > 0$).

The parametric form of the GEV encompasses that of the Gumbel, Frechet and reverse Weibull distributions, which are obtained for $s = 0$, $s > 0$ and $s < 0$ respectively. It was first introduced by Jenkinson (1955).

Value

dgev gives the density function, pgev gives the distribution function, qgev gives the quantile function, and rgev generates random deviates.

References

Jenkinson, A. F. (1955) The frequency distribution of the annual maximum (or minimum) of meteorological elements. *Quart. J. R. Met. Soc.*, **81**, 158–171.

See Also

fgev, rfrechet, rgumbel, rrweibull

Examples

```
dgev(2:4, 1, 0.5, 0.8)
pgev(2:4, 1, 0.5, 0.8)
qgev(seq(0.9, 0.6, -0.1), 2, 0.5, 0.8)
rgev(6, 1, 0.5, 0.8)
p <- (1:9)/10
pgev(qgev(p, 1, 2, 0.8), 1, 2, 0.8)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

---

gpd *The Generalized Pareto Distribution*

Description

Density function, distribution function, quantile function and random generation for the generalized Pareto distribution (GPD) with location, scale and shape parameters.

Usage

```
dgpd(x, loc=0, scale=1, shape=0, log = FALSE)
pgpd(q, loc=0, scale=1, shape=0, lower.tail = TRUE)
qgpd(p, loc=0, scale=1, shape=0, lower.tail = TRUE)
rgpd(n, loc=0, scale=1, shape=0)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
loc, scale, shape	Location, scale and shape parameters; the shape argument cannot be a vector (must have length one).
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The generalized Pareto distribution function (Pickands, 1975) with parameters loc $= a$, scale $= b$ and shape $= s$ is

$$G(z) = 1 - \{1 + s(z - a)/b\}^{-1/s}$$

for $1 + s(z - a)/b > 0$ and $z > a$, where $b > 0$. If $s = 0$ the distribution is defined by continuity.

Value

dgpd gives the density function, pgpd gives the distribution function, qgpd gives the quantile function, and rgpd generates random deviates.

References

Pickands, J. (1975) Statistical inference using extreme order statistics. *Annals of Statistics*, **3**, 119–131.

See Also

fpot, rgev

Examples

```
dgpd(2:4, 1, 0.5, 0.8)
pgpd(2:4, 1, 0.5, 0.8)
qgpd(seq(0.9, 0.6, -0.1), 2, 0.5, 0.8)
rgpd(6, 1, 0.5, 0.8)
p <- (1:9)/10
pgpd(qgpd(p, 1, 2, 0.8), 1, 2, 0.8)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

gumbel *The Gumbel Distribution*

Description

Density function, distribution function, quantile function and random generation for the Gumbel distribution with location and scale parameters.

Usage

```
dgumbel(x, loc=0, scale=1, log = FALSE)
pgumbel(q, loc=0, scale=1, lower.tail = TRUE)
qgumbel(p, loc=0, scale=1, lower.tail = TRUE)
rgumbel(n, loc=0, scale=1)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
loc, scale	Location and scale parameters (can be given as vectors).
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The Gumbel distribution function with parameters loc $= a$ and scale $= b$ is

$$G(z) = \exp\left\{-\exp\left[-\left(\frac{z-a}{b}\right)\right]\right\}$$

for all real z, where $b > 0$.

Value

dgumbel gives the density function, pgumbel gives the distribution function, qgumbel gives the quantile function, and rgumbel generates random deviates.

See Also

rfrechet, rgev, rrweibull

Examples

```
dgumbel(-1:2, -1, 0.5)
pgumbel(-1:2, -1, 0.5)
qgumbel(seq(0.9, 0.6, -0.1), 2, 0.5)
rgumbel(6, -1, 0.5)
p <- (1:9)/10
pgumbel(qgumbel(p, -1, 2), -1, 2)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

---

gumbelx *Maxima of Two Gumbel Distributions*

Description

Density function, distribution function, quantile function and random generation for the maxima of two Gumbel distributions, each with different location and scale parameters.

Usage

```
dgumbelx(x, loc1=0, scale1=1, loc2=0, scale2=1, log = FALSE)
pgumbelx(q, loc1=0, scale1=1, loc2=0, scale2=1, lower.tail = TRUE)
qgumbelx(p, interval, loc1=0, scale1=1, loc2=0, scale2=1, lower.tail = TRUE, ...)
rgumbelx(n, loc1=0, scale1=1, loc2=0, scale2=1)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
interval	A length two vector containing the end-points of the interval to be searched for the quantiles, passed to the uniroot function.
loc1, scale1, loc2, scale2	Location and scale parameters of the two Gumbel distributions. The second location parameter must be greater than or equal to the first location parameter.
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]
...	Other arguments passed to uniroot.

Value

dgumbelx gives the density function, pgumbelx gives the distribution function, qgumbelx gives the quantile function, and rgumbelx generates random deviates.

See Also

fgev, rfrechet, rgumbel, rrweibull, uniroot

Examples

```
dgumbelx(2:4, 0, 1.1, 1, 0.5)
pgumbelx(2:4, 0, 1.1, 1, 0.5)
qgumbelx(seq(0.9, 0.6, -0.1), interval = c(0,10), 0, 1.2, 2, 0.5)
rgumbelx(6, 0, 1.1, 1, 0.5)
p <- (1:9)/10
pgumbelx(qgumbelx(p, interval = c(0,10), 0, 0.5, 1, 2), 0, 0.5, 1, 2)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

---

hbvevd — *Parametric Spectral Density Functions of Bivariate Extreme Value Models*

Description

Calculate or plot the density h of the spectral measure H on the interval $(0, 1)$, for nine parametric bivariate extreme value models.

Usage

```
hbvevd(x = 0.5, dep, asy = c(1,1), alpha, beta, model = c("log", "alog",
    "hr", "neglog", "aneglog", "bilog", "negbilog", "ct", "amix"),
     half = FALSE, plot = FALSE, add = FALSE, lty = 1, ...)
```

Arguments

x	A vector of values at which the function is evaluated (ignored if plot or add is TRUE). $h(1/2)$ is returned by default.
dep	Dependence parameter for the logistic, asymmetric logistic, Husler-Reiss, negative logistic and asymmetric negative logistic models.
asy	A vector of length two, containing the two asymmetry parameters for the asymmetric logistic and asymmetric negative logistic models.
alpha, beta	Alpha and beta parameters for the bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models.
model	The specified model; a character string. Must be either "log" (the default), "alog", "hr", "neglog", "aneglog", "bilog", "negbilog", "ct" or "amix" (or any unique partial match), for the logistic, asymmetric logistic, Husler-Reiss, negative logistic, asymmetric negative logistic, bilogistic, negative bilogistic, Coles-Tawn and asymmetric mixed models respectively. The definition of each model is given in rbvevd. If parameter arguments are given that do not correspond to the specified model those arguments are ignored, with a warning.
half	Logical; if TRUE the function is divided by two, corresponding to a spectral measure with total mass one rather than two.
plot	Logical; if TRUE the function is plotted. The x and y values used to create the plot are returned invisibly.

add	Logical; add to an existing plot?
lty	Line type.
...	Other high-level graphics parameters to be passed to plot.

Details

Any bivariate extreme value distribution can be written as

$$G(z_1, z_2) = \exp\left[-\int_0^1 \max\{wy_1, (1-w)y_2\}H(dw)\right]$$

for some function $H(\cdot)$ defined on $[0, 1]$, satisfying

$$\int_0^1 wH(dw) = \int_0^1 (1-w)H(dw) = 1$$

In particular, the total mass of H is two. The functions y_1 and y_2 are as defined in abvevd.

H is called the spectral measure, with density h on the interval $(0, 1)$.

Value

hbvevd calculates or plots the spectral density function h for one of nine parametric bivariate extreme value models, at specified parameter values.

Point Masses

For differentiable models H may have up to two point masses: at zero and one. Assuming that the model parameters are in the interior of the parameter space, we have the following. For the asymmetric logistic and asymmetric negative logistic models the point masses are of size 1-asy1 and 1-asy2 respectively. For the asymmetric mixed model they are of size 1-alpha-beta and 1-alpha-2*beta respectively. For all other models the point masses are zero.

At independence, H has point masses of size one at both zero and one. At complete dependence [a non-differentiable model] H has a single point mass of size two at $1/2$. In either case, h is zero everywhere.

See Also

abvevd, fbvevd, rbvevd, plot.bvevd

Examples

```
hbvevd(dep = 2.7, model = "hr")
hbvevd(seq(0.25,0.5,0.75), dep = 0.3, asy = c(.7,.9), model = "alog")
hbvevd(alpha = 0.3, beta = 1.2, model = "negbi", plot = TRUE)

bvdata <- rbvevd(100, dep = 0.7, model = "log")
M1 <- fitted(fbvevd(bvdata, model = "log"))
hbvevd(dep = M1["dep"], model = "log", plot = TRUE)
```

lisbon *Annual Maximum Wind Speeds at Lisbon*

Description

A numeric vector containing annual maximum wind speeds, in kilometers per hour, from 1941 to 1970 at Lisbon, Portugal.

Usage

```
lisbon
```

Format

A vector containing 30 observations.

Source

Tiago de Oliveira, J. (1997) *Statistical Analysis of Extremes.* Pendor.

lossalae *General Liability Claims*

Description

The `lossalae` data frame has 1500 rows and 2 columns. The columns contain the indemnity payment (loss), and the allocated loss adjustment expense (alae), both in USD. The latter is the additional expenses associated with the settlement of the claim (e.g. claims investigation expenses and legal fees).

The dataset also has an attribute called `capped`, which gives the row names of the indemnity payments that were capped at their policy limit.

Usage

```
lossalae
```

Format

This data frame contains the following columns:

Loss A numeric vector containing the indemnity payments.

ALAE A numeric vector containing the allocated loss adjustment expenses.

Source

Frees, E. W. and Valdez, E. A. (1998) Understanding relationships using copulas. *North American Actuarial Journal*, **2**, 1–15.

References

Klugman, S. A. and Parsa, R. (1999) Fitting bivariate loss distributions with copulas. *Insurance: Mathematics and Economics*, **24**, 139–148.

Beirlant, J., Goegebeur, Y., Segers, J. and Teugels, J. L. (2004) *Statistics of Extremes: Theory and Applications.*, Chichester, England: John Wiley and Sons.

`marma` *Simulate MARMA(p,q) Processes*

Description

Simulation of MARMA(p,q) processes.

Usage

```
marma(n, p = 0, q = 0, psi, theta, init = rep(0, p), n.start = p,
    rand.gen = rfrechet, ...)
mar(n, p = 1, psi, init = rep(0, p), n.start = p, rand.gen =
    rfrechet, ...)
mma(n, q = 1, theta, rand.gen = rfrechet, ...)
```

Arguments

`n`	The number of observations.
`p`	The AR order of the MARMA process.
`q`	The MA order of the MARMA process.
`psi`	A vector of non-negative parameters, of length `p`. Can be omitted if `p` is zero.
`theta`	A vector of non-negative parameters, of length `q`. Can be omitted if `q` is zero.
`init`	A vector of non-negative starting values, of length `p`.
`n.start`	A non-negative value denoting the length of the burn-in period. If `n.start` is less than `p`, then `p` minus `n.start` starting values will be included in the output series.
`rand.gen`	A simulation function to generate the innovations.
`...`	Additional arguments for `rand.gen`. Most usefully, the scale and shape parameters of the innovations generated by `rfrechet` can be specified by `scale` and `shape` respectively.

Details

A max autoregressive moving average process $\{X_k\}$, denoted by MARMA(p,q), is defined in Davis and Resnick (1989) as satisfying

$$X_k = \max\{\phi_1 X_{k-1}, \ldots, \phi_p X_{k-p}, \epsilon_k, \theta_1 \epsilon_{k-1}, \ldots, \theta_q \epsilon_{k-q}\}$$

where phi $= (\phi_1, \ldots, \phi_p)$ and theta $= (\theta_1, \ldots, \theta_q)$ are non-negative vectors of parameters, and where $\{\epsilon_k\}$ is a series of *iid* random variables with a common distribution defined by rand.gen.

The functions mar and mma generate MAR(p) and MMA(q) processes respectively. A MAR(p) process $\{X_k\}$ is equivalent to a MARMA(p, 0) process, so that

$$X_k = \max\{\phi_1 X_{k-1}, \ldots, \phi_p X_{k-p}, \epsilon_k\}.$$

A MMA(q) process $\{X_k\}$ is equivalent to a MARMA(0, q) process, so that

$$X_k = \max\{\epsilon_k, \theta_1 \epsilon_{k-1}, \ldots, \theta_q \epsilon_{k-q}\}.$$

Value

A numeric vector of length n.

References

Davis, R. A. and Resnick, S. I. (1989) Basic properties and prediction of max-arma processes. *Adv. Appl. Prob.*, **21**, 781–803.

See Also

evmc

Examples

```
marma(100, p = 1, q = 1, psi = 0.75, theta = 0.65)
mar(100, psi = 0.85, n.start = 20)
mma(100, q = 2, theta = c(0.75, 0.8))
```

---

mrlplot *Empirical Mean Residual Life Plot*

Description

The empirical mean residual life plot.

Usage

```
mrlplot(data, tlim, pscale = FALSE, nt = max(100, length(data)), lty =
    c(2,1,2), col = 1, conf = 0.95, main = "Mean Residual Life Plot",
    xlab = "Threshold", ylab = "Mean Excess", ...)
```

Arguments

data	A numeric vector.
tlim	A numeric vector of length two, giving the limits for the thresholds at which the mean residual life plot is evaluated. If tlim is not given, sensible defaults are used.
pscale	If TRUE, then the x-axis gives the threshold exceedance probability rather than the threshold itself.
nt	The number of thresholds at which the mean residual life plot is evaluated.
lty, col	Arguments passed to matplot. The first and last elements of lty correspond to the lower and upper confidence limits respectively. Use zero to supress.
conf	The (pointwise) confidence coefficient for the plotted confidence intervals.
main	Plot title.
xlab, ylab	x and y axis labels.
...	Other arguments to be passed to matplot.

Details

The empirical mean residual life plot is the locus of points

$$\left(u, \frac{1}{n_u} \sum_{i=1}^{n_u} (x_{(i)} - u) \right)$$

where $x_{(1)}, \ldots, x_{(n_u)}$ are the n_u observations that exceed the threshold u. If the exceedances of a threshold u_0 are generalized Pareto, the empirical mean residual life plot should be approximately linear for $u > u_0$.

The confidence intervals within the plot are symmetric intervals based on the approximate normality of sample means.

Value

A list with components x and y is invisibly returned. The components contain those objects that were passed to the formal arguments x and y of matplot in order to create the mean residual life plot.

Author(s)

Stuart Coles and Alec Stephenson

See Also

fpot, matplot, tcplot

Examples

```
mrlplot(portpirie)
```

`mtransform` *GEV Transformations*

Description

Transforms to exponential margins under the GEV model.

Usage

```
mtransform(x, p, inv = FALSE, drp = FALSE)
```

Arguments

`x`	A matrix with n rows and d columns, or a vector. In the latter case, if `p` is a list with the same length as the vector, it is treated as a matrix with one row. If `p` is not a list, it is treated as a matrix with one column.
`p`	A vector of length three or a matrix with n rows and three columns. It can also be a list of length d, in which case each element must be a vector of length three or a matrix with n rows and three columns.
`inv`	Logical; use the inverse transformation?
`drp`	Logical; return a vector rather than a single row matrix?. Note that a single column matrix is always returned as a vector.

Details

Let x_i denote a vector of observations for $i = 1, \ldots, n$. This function implements the transformation

$$y_i = \{1 + s_i(x_i - a_i)/b_i\}_+^{-1/s_i}$$

to each column of the matrix `x`.

The values (a_i, b_i, s_i) are contained in the ith row of the n by 3 matrix `p`. If `p` is a vector of length three, the parameters are the same for every $i = 1, \ldots, n$. Alternatively, `p` can be a list with `d` elements, in which case the jth element is used to transform the jth column of `x`.

This function is mainly for internal use. It is used by bivariate and multivariate routines to calculate marginal transformations.

Value

A numeric matrix or vector.

mvevd *Parametric Multivariate Extreme Value Distributions*

Description

Density function, distribution function and random generation for the multivariate logistic and multivariate asymmetric logistic models.

Usage

```
pmvevd(q, dep, asy, model = c("log", "alog"), d = 2, mar = c(0,1,0),
    lower.tail = TRUE)
rmvevd(n, dep, asy, model = c("log", "alog"), d = 2, mar = c(0,1,0))
dmvevd(x, dep, asy, model = c("log", "alog"), d = 2, mar = c(0,1,0),
    log = FALSE)
```

Arguments

x, q	A vector of length d or a matrix with d columns, in which case the density/distribution is evaluated across the rows.
n	Number of observations.
dep	The dependence parameter(s). For the logistic model, should be a single value. For the asymmetric logistic model, should be a vector of length $2^d - d - 1$, or a single value, in which case the value is used for each of the $2^d - d - 1$ parameters (see **Details**).
asy	The asymmetry parameters for the asymmetric logistic model. Should be a list with $2^d - 1$ vector elements containing the asymmetry parameters for each separate component (see **Details**).
model	The specified model; a character string. Must be either "log" (the default) or "alog" (or any unique partial match), for the logistic and asymmetric logistic models respectively.
d	The dimension.
mar	A vector of length three containing marginal parameters for every univariate margin, or a matrix with three columns where each column represents a vector of values to be passed to the corresponding marginal parameter. It can also be a list with d elements, such that each element is either a vector of length three or a matrix with three columns, in which case the ith element represents the marginal parameters on the ith margin.
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default), the distribution function is returned; the survivor function is returned otherwise.

Details

Define

$$y_i = y_i(z_i) = \{1 + s_i(z_i - a_i)/b_i\}^{-1/s_i}$$

for $1 + s_i(z_i - a_i)/b_i > 0$ and $i = 1, \ldots, d$, where the marginal parameters are given by (a_i, b_i, s_i), $b_i > 0$. If $s_i = 0$ then y_i is defined by continuity. Let $z = (z_1, z_2, \ldots, z_d)$. In each of the multivariate distributions functions $G(z)$ given below, the univariate margins are generalized extreme value, so that $G(z_i) = \exp(-y_i)$ for $i = 1, \ldots, d$. If $1 + s_i(z_i - a_i)/b_i \leq 0$ for some $i = 1, \ldots, d$, the value z_i is either greater than the upper end point (if $s_i < 0$), or less than the lower end point (if $s_i > 0$), of the ith univariate marginal distribution.

`model = "log"` (Gumbel, 1960)

The `d` dimensional multivariate logistic distribution function with parameter `dep` $= r$ is

$$G(z) = \exp\left\{-\left(\sum\nolimits_{i=1}^{d} y_i^{1/r}\right)^r\right\}$$

where $0 < r \leq 1$. This is a special case of the multivariate asymmetric logistic model.

`model = "alog"` (Tawn, 1990)

Let B be the set of all non-empty subsets of $\{1, \ldots, d\}$, let $B_1 = \{b \in B : |b| = 1\}$, where $|b|$ denotes the number of elements in the set b, and let $B_{(i)} = \{b \in B : i \in b\}$. The `d` dimensional multivariate asymmetric logistic distribution function is

$$G(z) = \exp\left\{-\sum\nolimits_{b \in B}\left[\sum\nolimits_{i \in b}(t_{i,b}y_i)^{1/r_b}\right]^{r_b}\right\},$$

where the dependence parameters $r_b \in (0, 1]$ for all $b \in B \setminus B_1$, and the asymmetry parameters $t_{i,b} \in [0, 1]$ for all $b \in B$ and $i \in b$. The constraints $\sum_{b \in B_{(i)}} t_{i,b} = 1$ for $i = 1, \ldots, d$ ensure that the marginal distributions are generalized extreme value. Further constraints arise from the possible redundancy of asymmetry parameters in the expansion of the distribution form. Let $b_{-i_0} = \{i \in b : i \neq i_0\}$. If $r_b = 1$ for some $b \in B \setminus B_1$ then $t_{i,b} = 0$ for all $i \in b$. Furthermore, if for some $b \in B \setminus B_1$, $t_{i,b} = 0$ for all $i \in b_{-i_0}$, then $t_{i_0,b} = 0$.

`dep` should be a vector of length $2^d - d - 1$ which contains $\{r_b : b \in B \setminus B_1\}$, with the order defined by the natural set ordering on the index. For example, for the trivariate model, `dep` $= (r_{12}, r_{13}, r_{23}, r_{123})$. `asy` should be a list with $2^d - 1$ elements. Each element is a vector which corresponds to a set $b \in B$, containing $t_{i,b}$ for every integer $i \in b$. The elements should be given using the natural set ordering on the $b \in B$, so that the first d elements are vectors of length one corresponding to the sets $\{1\}, \ldots, \{d\}$, and the last element is a a vector of length d, corresponding to the set $\{1, \ldots, d\}$. `asy` must be constructed to ensure that all constraints are satisfied or an error will occur.

Value

`pmvevd` gives the distribution function, `dmvevd` gives the density function and `rmvevd` generates random deviates, for the multivariate logistic or multivariate asymmetric logistic model.

Note

Multivariate extensions of other bivariate models are more complex. A multivariate extension of the Husler-Reiss model exists, involving a multidimensional integral and one parameter for each

bivariate margin. Multivariate extensions for the negative logistic model can be derived but are considerably more complex and appear to be less flexible. The "multivariate negative logistic model" often presented in the literature (e.g. Kotz *et al*, 2000) is not a valid distribution function and should not be used.

The logistic and asymmetric logistic models respectively are simulated using Algorithms 2.1 and 2.2 in Stephenson(2003b).

The density function of the logistic model is evaluated using the representation of Shi(1995). The density function of the asymmetric logistic model is evaluated using the representation given in Stephenson(2003a).

References

Gumbel, E. J. (1960) Distributions des valeurs extremes en plusieurs dimensions. *Publ. Inst. Statist. Univ. Paris*, **9**, 171–173.

Kotz, S. and Balakrishnan, N. and Johnson, N. L. (2000) *Continuous Multivariate Distributions*, vol. 1. New York: John Wiley & Sons, 2nd edn.

Shi, D. (1995) Fisher information for a multivariate extreme value distribution. *Biometrika*, **82**(3), 644–649.

Stephenson, A. G. (2003a) *Extreme Value Distributions and their Application*. Ph.D. Thesis, Lancaster University, Lancaster, UK.

Stephenson, A. G. (2003b) Simulating multivariate extreme value distributions of logistic type. *Extremes*, **6**(1), 49–60.

Tawn, J. A. (1990) Modelling multivariate extreme value distributions. *Biometrika*, **77**, 245–253.

See Also

rbvevd, rgev

Examples

```
pmvevd(matrix(rep(0:4,5), ncol=5), dep = .7, model = "log", d = 5)
pmvevd(rep(4,5), dep = .7, model = "log", d = 5)
rmvevd(10, dep = .7, model = "log", d = 5)
dmvevd(rep(-1,20), dep = .7, model = "log", d = 20, log = TRUE)

asy <- list(.4, .1, .6, c(.3,.2), c(.1,.1), c(.4,.1), c(.2,.3,.2))
pmvevd(rep(2,3), dep = c(.6,.5,.8,.3), asy = asy, model = "alog", d = 3)
asy <- list(.4, .0, .6, c(.3,.2), c(.1,.1), c(.4,.1), c(.2,.4,.2))
rmvevd(10, dep = c(.6,.5,.8,.3), asy = asy, model = "alog", d = 3)
dmvevd(rep(0,3), dep = c(.6,.5,.8,.3), asy = asy, model = "alog", d = 3)

asy <- list(0, 0, 0, 0, c(0,0), c(0,0), c(0,0), c(0,0), c(0,0), c(0,0),
  c(.2,.1,.2), c(.1,.1,.2), c(.3,.4,.1), c(.2,.2,.2), c(.4,.6,.2,.5))
rmvevd(10, dep = .7, asy = asy, model = "alog", d = 4)
rmvevd(10, dep = c(rep(1,6), rep(.7,5)), asy = asy, model = "alog", d = 4)
```

ocmulgee *Maximum Annual Flood Discharges of the Ocmulgee River*

Description

The ocmulgee data frame has 40 rows and 2 columns. The columns contain maximum annual flood discharges, in units of 1000 cubed feet per second, from the Ocmulgee River in Georgia, USA at Hawkinsville (upstream) and Macon (downstream), for the years 1910 to 1949. The row names give the years of observation.

Usage

```
ocmulgee
```

Format

This data frame contains the following columns:

hawk A numeric vector containing maximum annual flood discharges at Hawkinsville (upstream).

macon A numeric vector containing maximum annual flood discharges at Macon (downstream).

Source

Gumbel, E. J. and Goldstein, N. (1964) Analysis of empirical bivariate extremal distributions. *J. Amer. Statist. Assoc.*, **59**, 794–816.

oldage *Oldest Ages for Swedish Males and Females*

Description

The oldage data frame has 66 rows and 2 columns. The columns contain the oldest ages at death for men and women in Sweden, for the period 1905–1970. The row names give the years of observation.

Usage

```
oldage
```

Format

This data frame contains the following columns:

men A numeric vector containing the oldest ages at death for men.

women A numeric vector containing the oldest ages at death for women.

Source

Fransen, A. and Tiago de Oliveira, J. (1984) Statistical choice of univariate extreme models, part II, in *Statistical Extremes and Applications*, J. Tiago de Oliveira ed., 373–394, D. Reidel, Dordrect.

`order` *Distributions of Order Statistics*

Description

Density function, distribution function and random generation for a selected order statistic of a given number of independent variables from a specified distribution.

Usage

```
dorder(x, densfun, distnfun, ..., distn, mlen = 1, j = 1,
      largest = TRUE, log = FALSE)
porder(q, distnfun, ..., distn, mlen = 1, j = 1, largest = TRUE,
      lower.tail = TRUE)
rorder(n, quantfun, ..., distn, mlen = 1, j = 1, largest = TRUE)
```

Arguments

`x`, `q`	Vector of quantiles.
`n`	Number of observations.
`densfun`, `distnfun`, `quantfun`	Density, distribution and quantile function of the specified distribution. The density function must have a `log` argument (a simple wrapper can always be constructed to achieve this).
`...`	Parameters of the specified distribution.
`distn`	A character string, optionally specified as an alternative to `densfun`, `distnfun` and `quantfun` such that the density, distribution and quantile functions are formed upon the addition of the prefixes `d`, `p` and `q` respectively.
`mlen`	The number of independent variables.
`j`	The order statistic, taken as the `j`th largest (default) or smallest of `mlen`, according to the value of `largest`.
`largest`	Logical; if `TRUE` (default) use the `j`th largest order statistic, otherwise use the `j`th smallest.
`log`	Logical; if `TRUE`, the log density is returned.
`lower.tail`	Logical; if `TRUE` (default) probabilities are P[X <= x], otherwise P[X > x].

Value

`dorder` gives the density function, `porder` gives the distribution function and `qorder` gives the quantile function of a selected order statistic from a sample of size `mlen`, from a specified distibution. `rorder` generates random deviates.

See Also

rextreme, rgev

Examples

```
dorder(2:4, dnorm, pnorm, mean = 0.5, sd = 1.2, mlen = 5, j = 2)
dorder(2:4, distn = "norm", mean = 0.5, sd = 1.2, mlen = 5, j = 2)
dorder(2:4, distn = "exp", mlen = 2, j = 2)
porder(2:4, distn = "exp", rate = 1.2, mlen = 2, j = 2)
rorder(5, qgamma, shape = 1, mlen = 10, j = 2)
```

---

oxford *Annual Maximum Temperatures at Oxford*

Description

A numeric vector containing annual maximum temperatures, in degrees Fahrenheit, from 1901 to 1980 at Oxford, England.

Usage

```
oxford
```

Format

A vector containing 80 observations.

Source

Tabony, R. C. (1983) Extreme value analysis in meteorology. *The Meteorological Magazine* **112**, 77–98.

---

plot.bvevd *Plot Diagnostics for a Bivariate EVD Object*

Description

Six plots (selectable by `which`) are currently provided: two conditional P-P plots (1,2), conditioning on each margin, a density plot (3), a dependence function plot (4), a quantile curves plot (5) and a spectral density plot (6). Plot diagnostics for the generalized extreme value margins (selectable by `mar` and `which`) are also available.

Usage

```
## S3 method for class 'bvevd'
plot(x, mar = 0, which = 1:6, main, ask = nb.fig <
    length(which) && dev.interactive(), ci = TRUE, cilwd = 1,
    a = 0, grid = 50, legend = TRUE, nplty = 2, blty = 3, method = "cfg",
    convex = FALSE, rev = FALSE, p = seq(0.75, 0.95, 0.05),
    mint = 1, half = FALSE, ...)
```

Arguments

x	An object of class "bvevd".
mar	If mar = 1 or mar = 2 diagnostics are given for the first or second genereralized extreme value margin respectively.
which	A subset of the numbers 1:6 selecting the plots to be shown. By default all are plotted.
main	Title of each plot. If given, should be a character vector with the same length as which.
ask	Logical; if TRUE, the user is asked before each plot.
ci	Logical; if TRUE (the default), plot simulated 95% confidence intervals for the conditional P-P plots.
cilwd	Line width for confidence interval lines.
a	Passed through to ppoints for empirical estimation. Larger values give less probability for extreme events.
grid	Argument for the density plot. The (possibly transformed) data is plotted with a contour plot of the bivariate density of the fitted model. The density is evaluated at grid^2 points.
legend	If legend is TRUE and if the fitted data contained a third column of mode logical, then a legend is included in the density and quantile curve plots.
method, convex, rev	Arguments to the dependence function plot. The dependence function for the fitted model is plotted and (optionally) compared to a non-parameteric estimate. See abvnonpar for a description of the arguments.
nplty, blty	Line types for the dependence function plot. nplty is the line type of the non-parametric estimate. To omit the non-parametric estimate set nplty to zero. blty is the line type of the triangular border. To omit the border estimate set blty to zero.
p, mint	Arguments to the quantile curves plot. See qcbvnonpar for a description of the plot and the arguments.
half	Argument to the spectral density plot. See hbvevd.
...	Other arguments to be passed through to plotting functions.

Details

In all plots we assume that the fitted model is stationary. For non-stationary models the data are transformed to stationarity. The plot then corresponds to the distribution obtained when all covariates are zero. In particular, the density and quanitle curves plots will not plot the original data for non-stationary models.

A conditional P-P plot is a P-P plot for the condition distribution function of a bivariate evd object. Let $G(.|.)$ be the conditional distribution of the first margin given the second, under the fitted model. Let $z_1, \ldots, z_m$ be the data used in the fitted model, where $z_j = (z_{1j}, z_{2j})$ for $j = 1, \ldots, m$. The plot that (by default) is labelled Conditional Plot Two, conditioning on the second margin, consists of the points

$$\{(p_i, c_i), i = 1, \ldots, m\}$$

where $p_1, \ldots, p_m$ are plotting points defined by `ppoints` and c_i is the ith largest value from the sample $\{G(z_{j1}|z_{j2}), j = 1, \ldots, m\}$. The margins are reversed for Conditional Plot One, so that $G(.|.)$ is the conditional distribution of the second margin given the first.

See Also

`plot.uvevd`, `contour`, `jitter`, `abvnonpar`, `qcbvnonpar`

Examples

```
bvdata <- rbvevd(100, dep = 0.6, model = "log")
M1 <- fbvevd(bvdata, model = "log")
## Not run: par(mfrow = c(2,2))
## Not run: plot(M1, which = 1:5)
## Not run: plot(M1, mar = 1)
## Not run: plot(M1, mar = 2)
```

`plot.bvpot` *Plot Diagnostics for a Bivariate POT EVD Object*

Description

Four plots (selectable by `which`) are currently provided: a density plot (1), a dependence function plot (2), a quantile curves plot (3) and a spectral density plot (4). Plot diagnostics for the generalized Pareto peaks-over-threshold margins (selectable by `mar` and `which`) are also available.

Usage

```
## S3 method for class 'bvpot'
plot(x, mar = 0, which = 1:4, main, ask = nb.fig <
    length(which) && dev.interactive(), grid = 50, above = FALSE,
    levels = NULL, tlty = 1, blty = 3, rev = FALSE, p = seq(0.75,
    0.95, 0.05), half = FALSE, ...)
```

Arguments

x	An object of class "bvpot".
mar	If mar = 1 or mar = 2 diagnostics are given for the first or second generalized Pareto margin respectively.
which	A subset of the numbers 1:4 selecting the plots to be shown. By default all are plotted.
main	Title of each plot. If given, should be a character vector with the same length as which.
ask	Logical; if TRUE, the user is asked before each plot.
grid, levels	Arguments for the density plot. The data is plotted with a contour plot of the bivariate density of the fitted model in the tail region. The density is evaluated at grid^2 points, and contours are plotted at the values given in the numeric vector levels. If levels is NULL (the default), the routine attempts to find sensible values.
above	Logical; if TRUE, only data points above both marginal thresholds are plotted.
tlty	Line type for the lines identifying the thresholds.
rev, blty	Arguments to the dependence function plot. See abvevd.
p	Lower tail probabilities for the quantile curves plot. The plot is of the same type as given by the function qcbvnonpar, but applied to the parametric bivariate threshold model.
half	Argument to the spectral density plot. See hbvevd.
...	Other arguments to be passed through to plotting functions.

See Also

plot.bvevd, contour, abvnonpar, qcbvnonpar, hbvevd

Examples

```
bvdata <- rbvevd(500, dep = 0.6, model = "log")
M1 <- fbvpot(bvdata, threshold = c(0,0), model = "log")
## Not run: plot(M1)
## Not run: plot(M1, mar = 1)
## Not run: plot(M1, mar = 2)
```

---

plot.profile.evd *Plot Profile Log-likelihoods*

Description

Displays profile log-likelihoods from a model profiled with profile.evd.

Usage

```
## S3 method for class 'profile.evd'
plot(x, which = names(x), main = NULL,
    ask = nb.fig < length(which) && dev.interactive(), ci = 0.95,
    clty = 2, ...)
```

Arguments

x	An object of class "profile.evd".
which	A character vector giving the parameters for which the profile deviance is plotted, and for which profile confidence intervals are calculated. By default all profiled parameters in x are used.
main	Title of each plot; a character vector, the same length as which.
ask	Logical; if TRUE, the user is asked before each plot.
ci	A numeric vector. For each parameter in which profile confidence intervals are calculated, for each confidence coefficient in ci (but see **Warning**). The intervals are returned invisibly as a list of vectors/matrices. Each plot then (by default) includes horizonal lines that represent each interval.
clty	The line type of the horizontal lines that represent the profile confidence intervals. To omit the lines set clty to zero.
...	Other graphics parameters.

Value

Profile devainces are plotted for each parameter in which. For calculation of profile confidence intervals, use the confint.profile.evd function.

Warning

The profile confidence intervals may not have confidence coefficient ci, because the usual asymptotic properties of maximum likelihood estimators may not hold. For the GEV model, the usual asymptotic properties hold when the shape parameter is greater than -0.5 (Smith, 1985).

References

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

See Also

confint.profile.evd, plot.profile2d.evd, profile.evd, profile2d.evd

Examples

```
uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
M1 <- fgev(uvdata)
## Not run: M1P <- profile(M1)
## Not run: par(mfrow = c(2,2))
```

```
## Not run: cint <- plot(M1P, ci = c(0.95, 0.99))
## Not run: cint
```

`plot.profile2d.evd` *Plot Joint Profile Log-likelihoods*

Description

Displays an image plot of the joint profile log-likelihood from a model profiled with `profile.evd` and `profile2d.evd`.

Usage

```
## S3 method for class 'profile2d.evd'
plot(x, main = NULL,
    ci = c(0.5, 0.8, 0.9, 0.95, 0.975, 0.99, 0.995),
    col = heat.colors(8), intpts = 75, xaxs = "r", yaxs = "r", ...)
```

Arguments

`x`	An object of class `"profile2d.evd"`.
`main`	Title of plot; a character string.
`ci`	A numeric vector whose length is one less than the length of `col`. The colours of the image plot, excluding the background colour, represent confidence sets with confidence coefficients `ci` (but see **Warning**).
`col`	A list of colors such as that generated by `rainbow`, `heat.colors`, `topo.colors`, `terrain.colors` or similar functions.
`intpts`	If the package **interp** is available, interpolation is performed using `intpts` points for each parameter. The function is interpolated at `intpts^2` points in total.
`xaxs`, `yaxs`	Graphics parameters (see `par`). The default, `"r"`, overrides the default set by `image`.
`...`	Other parameters to be passed to `image`.

Warning

The sets represented by different colours may not be confidence sets with confidence coefficients `ci`, because the usual asymptotic properties of maximum likelihood estimators may not hold. For the GEV model, the usual asymptotic properties hold when the shape parameter is greater than -0.5 (Smith, 1985).

References

Smith, R. L. (1985) Maximum likelihood estimation in a class of non-regular cases. *Biometrika*, **72**, 67–90.

See Also

plot.profile.evd, profile.evd, profile2d.evd

Examples

```
uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
M1 <- fgev(uvdata)
## Not run: M1P <- profile(M1)
## Not run: M1JP <- profile2d(M1, M1P, which = c("scale", "shape"))
## Not run: plot(M1JP)
```

---

plot.uvevd *Plot Diagnostics for a Univariate EVD Object*

Description

Four plots (selectable by `which`) are currently provided: a P-P plot, a Q-Q plot, a density plot and a return level plot.

Usage

```
## S3 method for class 'uvevd'
plot(x, which = 1:4, main, ask = nb.fig <
    length(which) && dev.interactive(), ci = TRUE, cilwd = 1,
    a = 0, adjust = 1, jitter = FALSE, nplty = 2, ...)
## S3 method for class 'gumbelx'
plot(x, interval, which = 1:4, main, ask = nb.fig <
    length(which) && dev.interactive(), ci = TRUE, cilwd = 1,
    a = 0, adjust = 1, jitter = FALSE, nplty = 2, ...)
```

Arguments

`x`	An object that inherits from class `"uvevd"`.
`which`	If a subset of the plots is required, specify a subset of the numbers `1:4`.
`main`	Title of each plot. If given, must be a character vector with the same length as `which`.
`ask`	Logical; if `TRUE`, the user is asked before each plot.
`ci`	Logical; if `TRUE` (the default), plot simulated 95% confidence intervals for the P-P, Q-Q and return level plots.
`cilwd`	Line width for confidence interval lines.
`a`	Passed through to `ppoints` for empirical estimation. Larger values give less probability for extreme events.

`adjust`, `jitter`, `nplty`
: Arguments to the density plot. The density of the fitted model is plotted with a rug plot and (optionally) a non-parameteric estimate. The argument `adjust` controls the smoothing bandwidth for the non-parametric estimate (see `density`). `jitter` is logical; if `TRUE`, the (possibly transformed) data are jittered to produce the rug plot. This need only be used if the data contains repeated values. `nplty` is the line type of the non-parametric estimate. To omit the non-parametric estimate set `nplty` to zero.

`interval`
: A vector of length two, for the gumbelx (maximum of two Gumbels) model. This is passed to the uniroot function to calculate quantiles for the Q-Q and return level plots. The interval should be large enough to contain all plotted quantiles or an error from uniroot will occur.

`...`
: Other parameters to be passed through to plotting functions.

Details

The following discussion assumes that the fitted model is stationary. For non-stationary generalized extreme value models the data are transformed to stationarity. The plot then corresponds to the distribution obtained when all covariates are zero.

The P-P plot consists of the points

$$\{(G_n(z_i), G(z_i)), i = 1, \ldots, m\}$$

where G_n is the empirical distribution function (defined using `ppoints`), G is the model based estimate of the distribution (generalized extreme value or generalized Pareto), and $z_1, \ldots, z_m$ are the data used in the fitted model, sorted into ascending order.

The Q-Q plot consists of the points

$$\{(G^{-1}(p_i), z_i), i = 1, \ldots, m\}$$

where G^{-1} is the model based estimate of the quantile function (generalized extreme value or generalized Pareto), $p_1, \ldots, p_m$ are plotting points defined by `ppoints`, and $z_1, \ldots, z_m$ are the data used in the fitted model, sorted into ascending order.

The return level plot for generalized extreme value models is defined as follows.

Let G be the generalized extreme value distribution function, with location, scale and shape parameters a, b and s respectively. Let z_t be defined by $G(z_t) = 1 - 1/t$. In common terminology, z_t is the return level associated with the return period t.

Let $y_t = -1/\log(1 - 1/t)$. It follows that

$$z_t = a + b(y_t^s - 1)/s.$$

When $s = 0$, z_t is defined by continuity, so that

$$z_t = a + b\log(y_t).$$

The curve within the return level plot is z_t plotted against y_t on a logarithmic scale, using maximum likelihood estimates of (a, b, s). If the estimate of s is zero, the curve will be linear. For large values of t, y_t is approximately equal to the return period t. It is usual practice to label the x-axis as the return period.

The points on the plot are

$$\{(-1/\log(p_i), z_i), i = 1, \ldots, m\}$$

where $p_1, \ldots, p_m$ are plotting points defined by ppoints, and $z_1, \ldots, z_m$ are the data used in the fitted model, sorted into ascending order. For a good fit the points should lie "close" to the curve.

The return level plot for peaks over threshold models is defined as follows.

Let G be the generalized Pareto distribution function, with location, scale and shape parameters u, b and s respectively, where u is the model threshold. Let z_m denote the m period return level (see fpot and the notation therein). It follows that

$$z_m = u + b((pmN)^s - 1)/s.$$

When $s = 0$, z_m is defined by continuity, so that

$$z_m = u + b\log(pmN).$$

The curve within the return level plot is z_m plotted against m on a logarithmic scale, using maximum likelihood estimates of (b, s, p). If the estimate of s is zero, the curve will be linear.

The points on the plot are

$$\{(1/(pN(1 - p_i)), z_i), i = 1, \ldots, m\}$$

where $p_1, \ldots, p_m$ are plotting points defined by ppoints, and $z_1, \ldots, z_m$ are the data used in the fitted model, sorted into ascending order. For a good fit the points should lie "close" to the curve.

See Also

plot.bvevd, density, jitter, rug, ppoints

Examples

```
uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
M1 <- fgev(uvdata)
## Not run: par(mfrow = c(2,2))
## Not run: plot(M1)

uvdata <- rgpd(100, loc = 0, scale = 1.1, shape = 0.2)
M1 <- fpot(uvdata, 1)
## Not run: par(mfrow = c(2,2))
## Not run: plot(M1)
```

---

portpirie *Annual Maximum Sea Levels at Port Pirie*

Description

A numeric vector containing annual maximum sea levels, in metres, from 1923 to 1987 at Port Pirie, South Australia.

Usage

```
portpirie
```

Format

A vector containing 65 observations.

Source

Tawn, J. A. (1993) Extreme sea-levels, in *Statistics in the Environment*, 243–263, eds. V. Barnett and F. Turkman, Wiley.

References

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*. London: Springer-Verlag.

---

profile.evd *Method for Profiling EVD Objects*

Description

Calculate profile traces for fitted models.

Usage

```
## S3 method for class 'evd'
profile(fitted, which = names(fitted$estimate), conf = 0.999,
    mesh = fitted$std.err[which]/4, xmin = rep(-Inf, length(which)),
    xmax = rep(Inf, length(which)), convergence = FALSE, method = "BFGS",
    control = list(maxit = 500), ...)
```

Arguments

fitted	An object of class "evd".
which	A character vector giving the model parameters that are to be profiled. By default, all parameters are profiled.
conf	Controls the range over which the parameters are profiled. The profile trace is constructed so that (assuming the usual asymptotic properties hold) profile confidence intervals with confidence coefficients conf or less can be derived from it.
mesh	A numeric vector containing one value for each parameter in which. The values represent the distance between the points profiled. By default mesh is one quarter of the standard errors. If the fitted object does not contain standard errors the argument must be specified. The argument should also be specified when an estimator is on or close to a parameter boundary, since the approximated "standard error" will then be close to zero.

xmin, xmax	Numeric vectors containing one value for each parameter in which. Each value represents the theoretical lower/upper bound of the corresponding parameter. The arguments are needed only when a parameter has a lower/upper bound at which the likelihood is non-zero. Do not use these arguments to specify plotting ranges in a subsequent plot (as they are used in the calculation of profile confidence intervals); to do this use xlim in the call to plot.
convergence	Logical; print convergence code after each optimization? (A warning is given for each non-zero convergence code, irrespective of the value of convergence.)
method	The optimization method.
control	Passed to optim. See optim for details.
...	Ignored.

Value

An object of class "profile.evd", which is a list with an element for each parameter being profiled. The elements are matrices. The first column contains the values of the profiled parameter. The second column contains profile deviances. The remaining columns contain the constrained maximum likelihood estimates for the remaining model parameters. For calculation of profile confidence intervals, use the confint.profile.evd function.

See Also

confint.profile.evd, profile2d.evd, plot.profile.evd

Examples

```
uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
M1 <- fgev(uvdata)
## Not run: M1P <- profile(M1)
## Not run: par(mfrow = c(2,2))
## Not run: cint <- plot(M1P)
## Not run: cint
```

---

profile2d.evd *Method for Profiling EVD Objects*

Description

Calculate joint profile traces for fitted models.

Usage

```
## S3 method for class 'evd'
profile2d(fitted, prof, which, pts = 20, convergence =
    FALSE, method = "Nelder-Mead", control = list(maxit = 5000), ...)
```

Arguments

fitted	An object of class "evd".
prof	An object of class "profile.evd", created using profile.evd with argument fitted. The object must contain the (marginal) profile traces for the two parameters specified in which.
which	A character vector of length two containing the original model parameters that are to be jointly profiled.
pts	The number of distinct values used for each profiled parameter in which. There are pts^2 optimizations performed in total.
convergence	Logical; print convergence code after each optimization? (A warning is given for each non-zero convergence code, irrespective of the value of convergence.)
method	The optimization method.
control	Passed to optim. See optim for details.
...	Ignored.

Value

An object of class "profile2d.evd", which is a list with three elements. The first element, a matrix named trace, has the same structure as the elements of an object of class "profile.evd". The last two elements give the distinct values used for each profiled parameter in which.

See Also

profile.evd, plot.profile2d.evd

Examples

```
uvdata <- rgev(100, loc = 0.13, scale = 1.1, shape = 0.2)
M1 <- fgev(uvdata)
## Not run: M1P <- profile(M1)
## Not run: M1JP <- profile2d(M1, M1P, which = c("scale", "shape"))
## Not run: plot(M1JP)
```

---

qcbvnonpar *Non-parametric Estimates for Bivariate Quantile Curves*

Description

Calculate or plot non-parametric estimates for quantile curves of bivariate extreme value distributions.

Usage

```
qcbvnonpar(p = seq(0.75, 0.95, 0.05), data, epmar = FALSE, nsloc1 =
    NULL, nsloc2 = NULL, mint = 1, method = c("cfg", "pickands",
    "tdo"), convex = FALSE, madj = 0, kmar = NULL, plot = FALSE,
    add = FALSE, lty = 1, lwd = 1, col = 1, xlim = range(data[,1],
    na.rm = TRUE), ylim = range(data[,2], na.rm = TRUE), xlab =
    colnames(data)[1], ylab = colnames(data)[2], ...)
```

Arguments

p	A vector of lower tail probabilities. One quantile curve is calculated or plotted for each probability.
data	A matrix or data frame with two columns, which may contain missing values.
epmar	If TRUE, an empirical transformation of the marginals is performed in preference to marginal parametric GEV estimation, and the nsloc arguments are ignored.
nsloc1, nsloc2	A data frame with the same number of rows as data, for linear modelling of the location parameter on the first/second margin. The data frames are treated as covariate matrices, excluding the intercept. A numeric vector can be given as an alternative to a single column data frame.
mint	An integer m. Quantile curves are plotted or calculated using the lower tail probabilities p^m.
method, kmar	Arguments for the non-parametric estimate of the dependence function. See abvnonpar.
convex, madj	Other arguments for the non-parametric estimate of the dependence function.
plot	Logical; if TRUE the data is plotted along with the quantile curves. If plot and add are FALSE (the default), the arguments following add are ignored.
add	Logical; add quantile curves to an existing data plot? The existing plot should have been created using either qcbvnonpar or plot.bvevd, the latter of which can plot quantile curves for parametric fits.
lty, lwd	Line types and widths.
col	Line colour.
xlim, ylim	x and y-axis limits.
xlab, ylab	x and y-axis labels.
...	Other high-level graphics parameters to be passed to plot.

Details

Let G be a fitted bivariate distribution function with margins G_1 and G_2. A quantile curve for a fitted distribution function G at lower tail probability p is defined by

$$Q(G, p) = \{(y_1, y_1) : G(y_1, y_2) = p\}.$$

For bivariate extreme value distributions, it consists of the points

$$\left\{G_1^{-1}(p_1), G_2^{-1}(p_2))\right\}$$

where $p_1 = p^{t/A(t)}$ and $p_2 = p^{(1-t)/A(t)}$, with A being the estimated dependence function defined in abvevd, and where t lies in the interval $[0, 1]$.

By default the margins G_1 and G_2 are modelled using estimated generalized extreme value distributions. For non-stationary generalized extreme value margins the plotted data are transformed to stationarity, and the plot corresponds to the distribution obtained when all covariates are zero.

If epmar is TRUE, empirical transformations are used in preference to generalized extreme value models. Note that the marginal empirical quantile functions are evaluated using quantile, which linearly interpolates between data points, hence the curve will not be a step function.

The idea behind the argument mint $= m$ is that if G is fitted to a dataset of componentwise maxima, and the underlying observations are *iid* distributed according to F, then if m is the size of the blocks over which the maxima were taken, approximately $F^m = G$, leading to $Q(F, p) = Q(G, p^m)$.

Value

qcbvnonpar calculates or plots non-parametric quantile curve estimates for bivariate extreme value distributions. If p has length one it returns a two column matrix giving points on the curve, else it returns a list of such matrices.

See Also

abvevd, abvnonpar, plot.bvevd

Examples

```
bvdata <- rbvevd(100, dep = 0.7, model = "log")
qcbvnonpar(c(0.9,0.95), data = bvdata, plot = TRUE)
qcbvnonpar(c(0.9,0.95), data = bvdata, epmar = TRUE, plot = TRUE)
```

---

rweibull *The Reverse Weibull Distribution*

Description

Density function, distribution function, quantile function and random generation for the reverse (or negative) Weibull distribution with location, scale and shape parameters.

Usage

```
drweibull(x, loc=0, scale=1, shape=1, log = FALSE)
prweibull(q, loc=0, scale=1, shape=1, lower.tail = TRUE)
qrweibull(p, loc=0, scale=1, shape=1, lower.tail = TRUE)
rrweibull(n, loc=0, scale=1, shape=1)

dnweibull(x, loc=0, scale=1, shape=1, log = FALSE)
pnweibull(q, loc=0, scale=1, shape=1, lower.tail = TRUE)
qnweibull(p, loc=0, scale=1, shape=1, lower.tail = TRUE)
rnweibull(n, loc=0, scale=1, shape=1)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
loc, scale, shape	Location, scale and shape parameters (can be given as vectors).
log	Logical; if TRUE, the log density is returned.
lower.tail	Logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

Details

The reverse (or negative) Weibull distribution function with parameters loc $= a$, scale $= b$ and shape $= s$ is

$$G(z) = \exp\left\{-\left[-\left(\frac{z-a}{b}\right)\right]^s\right\}$$

for $z < a$ and one otherwise, where $b > 0$ and $s > 0$.

Value

drweibull and dnweibull give the density function, prweibull and pnweibull give the distribution function, qrweibull and qnweibull give the quantile function, rrweibull and rnweibull generate random deviates.

Note

Within extreme value theory the reverse Weibull distibution (also known as the negative Weibull distribution) is often referred to as the Weibull distribution. We make a distinction to avoid confusion with the three-parameter distribution used in survival analysis, which is related by a change of sign to the distribution given above.

See Also

rfrechet, rgev, rgumbel

Examples

```
drweibull(-5:-3, -1, 0.5, 0.8)
prweibull(-5:-3, -1, 0.5, 0.8)
qrweibull(seq(0.9, 0.6, -0.1), 2, 0.5, 0.8)
rrweibull(6, -1, 0.5, 0.8)
p <- (1:9)/10
prweibull(qrweibull(p, -1, 2, 0.8), -1, 2, 0.8)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

sask *Maximum Annual Flood Discharges of the North Saskachevan River*

Description

A numeric vector containing maximum annual flood discharges, in units of 1000 cubic feet per second, of the North Saskachevan River at Edmonton, over a period of 47 years. Unfortunately, the data are ordered from largest to smallest.

Usage

```
sask
```

Format

A vector containing 47 observations.

Source

van Montfort, M. A. J. (1970) On testing that the distribution is of type I when type II is the alternative. *J. Hydrology*, **11**, 421–427.

sealevel *Annual Sea Level Maxima at Dover and Harwich*

Description

The `sealevel` data frame has 81 rows and 2 columns. The columns contain annual sea level maxima from 1912 to 1992 at Dover and Harwich respectively, two sites on the coast of Britain. The row names give the years of observation. There are 39 missing values.

Usage

```
sealevel
```

Format

This data frame contains the following columns:

dover A numeric vector containing annual sea level maxima at Dover, including 9 missing values.

harwich A numeric vector containing sea annual level maxima at Harwich, including 30 missing values.

Source

Coles, S. G. and Tawn, J. A. (1990) Statistics of coastal flood prevention. *Phil. Trans. R. Soc. Lond., A* **332**, 457–476.

sealevel2 *Annual Sea Level Maxima at Dover and Harwich with Indicator*

Description

The sealevel2 data frame has 81 rows and 3 columns. The first two columns contain annual sea level maxima from 1912 to 1992 at Dover and Harwich respectively, two sites on the coast of Britain. The third column is a logical vector denoting whether or not the maxima in a given year are assumed to have derived from the same storm event; this assumption is made if the times of obsevation of the maxima are at most 48 hours apart. The row names give the years of observation. There are 39 missing data values. There are only nine non-missing logical values.

Usage

```
sealevel2
```

Format

This data frame contains the following columns:

dover A numeric vector containing annual sea level maxima at Dover, including 9 missing values.

harwich A numeric vector containing sea annual level maxima at Harwich, including 30 missing values.

case A logical vector denoting whether or not the maxima are assumed to have derived from the same storm event.

Source

Coles, S. G. and Tawn, J. A. (1990) Statistics of coastal flood prevention. *Phil. Trans. R. Soc. Lond., A* **332**, 457–476.

tcplot *Threshold Choice Plot*

Description

Plots of parameter estimates at various thresholds for peaks over threshold modelling, using the Generalized Pareto or Point Process representation.

Usage

```
tcplot(data, tlim, model = c("gpd","pp"), pscale = FALSE, cmax =
    FALSE, r = 1, ulow = -Inf, rlow = 1, nt = 25, which = 1:npar,
    conf = 0.95, lty = 1, lwd = 1, type = "b", cilty = 1, vci =
    TRUE, xlab, xlim, ylabs, ylims, ask = nb.fig < length(which) &&
    dev.interactive(), ...)
```

Arguments

data	A numeric vector.
tlim	A numeric vector of length two, giving the limits for the thresholds at which the model is fitted.
model	The model; either "gpd" (the default) or "pp", for the Generalized Pareto or Point Process representations respectively.
pscale	If TRUE, then the x-axis gives the threshold exceedance probability rather than the threshold itself.
cmax	Logical; if FALSE (the default), the models are fitted using all exceedences over the thresholds. If TRUE, the models are fitted using cluster maxima, using clusters of exceedences derived from clusters.
r, ulow, rlow	Arguments used for the identification of clusters of exceedences (see clusters). Ignored if cmax is FALSE (the default).
nt	The number of thresholds at which the model is fitted.
which	If a subset of the plots is required, specify a subset of the numbers 1:npar, where npar is the number of parameters, so that npar = 2 when model = "gpd" (the default) and npar = 3 when model = "pp".
conf	The (pointwise) confidence coefficient for the plotted confidence intervals. Use zero to suppress.
lty, lwd	The line type and width of the line connecting the parameter estimates.
type	The form taken by the line connecting the parameter estimates and the points denoting these estimates. Possible values include "b" (the default) for points joined by lines, "o" for overplotted points and lines, and "l" for an unbroken line with no points.
cilty	The line type of the lines depicting the confidence intervals.
vci	If TRUE (the default), confidence intervals are plotted using vertical lines.
xlab, xlim	Label and limits for the x-axis; if given, these arguments apply to every plot.
ylabs, ylims	A vector of y-axis labels and a matrix of y-axis limits. If given, ylabs should have the same length as which, and ylims should have two columns and length(which) rows. If the length of which is one, then ylims can be a vector of length two.
ask	Logical; if TRUE, the user is asked before each plot.
...	Other arguments to be passed to the model fit function fpot.

Details

For each of the nt thresholds a peaks over threshold model is fitted using the function fpot. When model = "gpd" (the default), the maximum likelihood estimates for the shape and the modified scale parameter (modified by subtracting the shape multiplied by the threshold) are plotted against the thresholds. When model = "pp" the maximum likelihood estimates for the location, scale and shape parameters are plotted against the thresholds. (The modified scale parameter in the "gpd" case is equivalent to the scale parameter in the "pp" case.) If the threshold u is a valid threshold to be used for peaks over threshold modelling, the parameter estimates depicted should be approximately constant above u.

Value

A list is invisibly returned. Each component is a matrix with three columns giving parameter estimates and confidence limits.

Author(s)

Stuart Coles and Alec Stephenson

See Also

fpot, mrlplot, clusters

Examples

```
tlim <- c(3.6, 4.2)
## Not run: tcplot(portpirie, tlim)
## Not run: tcplot(portpirie, tlim, nt = 100, lwd = 3, type = "l")
## Not run: tcplot(portpirie, tlim, model = "pp")
```

---

uccle *Rainfall Maxima at Uccle, Belgium*

Description

The uccle data frame has 35 rows and 4 columns. The columns contain annual rainfall maxima (in millimetres) from 1938 to 1972 at Uccle, Belgium, over the durations of one day, one hour, ten minutes and one minute. The row names give the years of observation.

Usage

```
uccle
```

Format

This data frame contains the following columns:

day Annual daily rainfall maxima.

hour Annual hourly rainfall maxima.

tmin Annual rainfall maxima over ten minute durations.

min Annual rainfall maxima over one minute durations.

Source

Sneyers, R. (1977) L'intensite maximale des precipitations en Belgique. *Inst. Royal Meteor. Belgique, B* **86**.

venice *Largest Sea Levels in Venice*

Description

The venice data frame has 51 rows and 10 columns. The jth column contains the jth largest sea levels in Venice, for the years 1931–1981. Only the largest six measurements are available for the year 1935; the corresponding row contains four missing values. The years for each set of measurements are given as row names. A larger version of this data is available in the dataset venice2.

Usage

```
venice
```

Format

A data frame with 51 rows and 10 columns.

Source

Smith, R. L. (1986) Extreme value theory based on the r largest annual events. *Journal of Hydrology*, **86**, 27–43.

References

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*. London: Springer-Verlag.

venice2 *Largest Sea Levels in Venice*

Description

The venice2 data frame has 125 rows and 10 columns. The data was kindly provided by Anthony Davison. The jth column contains the jth largest sea levels in Venice, for the years 1887–2011. This is a larger version of the dataset venice. Only the largest six measurements are available for the year 1935, and only the largest is available for 1922; the corresponding rows contain missing values. The years for each set of measurements are given as row names.

Usage

```
venice2
```

Format

A data frame with 125 rows and 10 columns.

Source

Smith, R. L. (1986) Extreme value theory based on the r largest annual events. *Journal of Hydrology*, **86**, 27–43.

References

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*. London: Springer-Verlag.

Index